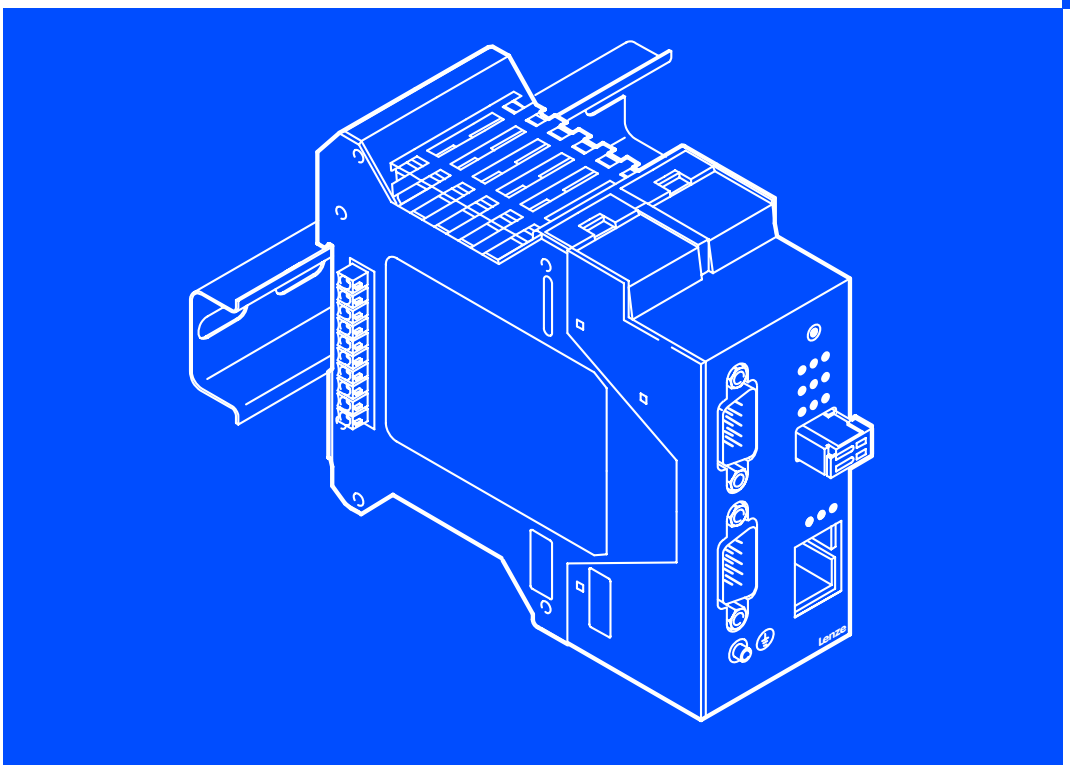
EDSTCXN
13144301

Softwarehandbuch

ETC Motion Control



ETCxCxxx

Betriebssystem NC

Lenze

Tipp!

Aktuelle Dokumentationen und Software-Updates zu Lenze Produkten finden Sie im Internet jeweils im Bereich ”Services & Downloads” unter

http://www.Lenze.com

© 2006 Lenze Drive Systems GmbH, Hans-Lenze-Straße 1, D-31855 Aerzen
Ohne besondere schriftliche Genehmigung von Lenze Drive Systems GmbH darf kein Teil dieser Dokumentation vervielfältigt oder Dritten zugänglich gemacht werden.
Wir haben alle Angaben in dieser Dokumentation mit größter Sorgfalt zusammengestellt und auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Trotzdem können wir Abweichungen nicht ganz ausschließen. Wir übernehmen keine juristische Verantwortung oder Haftung für Schäden, die dadurch eventuell entstehen. Notwendige Korrekturen werden wir in die nachfolgenden Auflagen einarbeiten.

1 Vorwort und Allgemeines 10

1.1 Über dieses Handbuch 10

1.2 Für welche Produkte ist das Handbuch gültig? 11

1.3 Rechtliche Bestimmungen 12

2 Erste Schritte 13

2.1 Systemübersicht 13

2.1.1 Beispiel für ein Automatisierungssystem 13

2.1.2 Beispiel für den Aufbau einer ETC-Insel 15

2.1.3 ETCHx und PC verbinden 16

2.2 Statusanzeige 17

2.3 Inbetriebnahmeschritte (Übersicht) 20

2.4 Kommunikation zwischen PC und ETCHx herstellen 21

2.4.1 ETCHx starten 21

2.4.2 Terminalprogramm ”HyperTerminal” starten und Monitorschnittstelle aktivieren 21

2.4.3 Betriebsart der ETCHx einstellen 25

2.4.4 IP-Adresse der ETCHx zuweisen 28

2.5 ETC-MMI und ETC-MMI-Gateway konfigurieren 30

2.6 ETC-MMI installieren 31

2.6.1 ETCPx einbauen und installieren 31

2.7 ETC-MMI starten 32

2.7.1 Sprache im ETC-MMI umschalten 33

2.7.2 Verbindung zwischen ETC-MMI und ETC herstellen 34

2.8 Antriebe über Maschinenkonstanten parametrieren 36

2.8.1 Übersicht der wichtigsten Maschinenkonstanten 36

2.8.2 Maschinenkonstanten-Datei ETCxC.mk 39

2.8.3 Hinweise zum Laden der MK-Datei in die Steuerung 40

2.8.4 Beispiel für das Anpassen einer Maschinenkonstanten-Datei 41

2.8.5 Maschinenkonstanten im ETC-MMI anpassen 42

2.8.6 Parameter der Antriebe überprüfen 43

2.8.7 Antriebe im Tippbetrieb testen 44

2.9 CNC-Programmierung nach DIN 66025 45

2.9.1 G-Funktionen 45

2.9.2 M-Funktionen 46

2.10 CNC-Beispielprogramm erstellen 47

2.10.1 Texteditor im ETC-MMI aufrufen 47

2.10.2 CNC-Programm eingeben und speichern 48

2.10.3 CNC-Programm in die Steuerung laden und starten 49

2.10.4 CNC-Programm erweitern 50

2.11 ETC-SPS-Programmierung mit CoDeSys 51
2.11.1 CoDeSys installieren 51
2.11.2 Steuerung im ETC-CoDeSys konfigurieren 51
2.12 SPS-Beispielprogramm erstellen 58
2.12.1 Vorausgesetzte Hardware 58
2.12.2 SPS-Beispielprogramm starten und konfigurieren 59
2.12.3 SPS-Beispielprogramm in die Steuerung laden 66
2.13 CNC- und SPS-Programm testen 70
2.14 SPS-Tasten im ETC-MMI 73
2.14.1 Beschriftung der SPS-Tasten im ETC-MMI 73
2.14.2 Aufruf der Signale in der SPS 74
2.14.3 Funktion der Tasten 74
2.15 Bedienung über ein Lenze-HMI 75
2.15.1 Einstellungen für den Anschluss eines Lenze-HMI H505 75
2.15.2 Funktionsbeschreibung HMI505-Bedienung 79
2.16 Firmware der ETCHx in der Betriebsart ”Stand-Alone” aktualisieren . . . 81
2.16.1 Aufruf des Bootmonitors in der Steuerung 81
2.16.2 Versionsabfrage der Firmware 82
2.16.3 Updaten der Firmware 82

3 CNC-Programmierung 85

3.1 Grundlagen 85
3.2 G-Funktionen 90
3.2.1 G-Funktionen im Überblick 90
3.2.2 G-Funktionen-Einzelbeschreibungen 93
3.3 Formelprozessor 170
3.3.1 Rechenoperationen 170
3.4 Satzerweiterungen 172
3.4.1 Parameterzuweisung P 172
3.4.2 H-Funktionen 173
3.4.3 M-Funktionen 173
3.4.4 Q-Funktionen 175
3.4.5 S-Funktionen 176
3.4.6 T-Funktionen 177
3.5 Datenfelder 178
3.5.1 P-Feld 178
3.5.2 Q-Feld 187

4 Maschinenkonstanten 188
4.1 Grundlagen 188
4.2 Testeinstellungen 190
4.2.1 MK_TEST_OHNEMECHANIK 190
4.2.2 MK_SPS_DUMMY 190
4.3 Software-Konfiguration 191
4.3.1 MK_KUNDE 191
4.3.2 MK_VOREINSTELLUNG 191
4.3.3 MK_NCPROG_OHNE_KOMMENTARE 192
4.3.4 MK_NCPROG_NICHT_INS_EEPROM 192
4.3.5 MK_METRISCH 192
4.3.6 MK_CONST_REL_MM 193
4.3.7 MK_CONST_REL_INCH 193
4.3.8 MK_KONTURFEHLER 193
4.3.9 MK_LAH_GRENZWINKEL 194
4.3.10 MK_RADIUS_B_BEWERTUNG 194
4.3.11 MK_EPSILONMM 195
4.3.12 MK_EPSILONGRAD 195
4.3.13 MK_OVERRIDEMAX 195
4.3.14 MK_FEHLERRESTART 195
4.3.15 MK_S0T0_VERSATZ_ERLAUBT 196
4.3.16 MK_S_VERSATZSPERRE 196
4.3.17 MK_CANOPEN_BAUDRATE 196
4.3.18 MK_DELTAT 196
4.4 Speicherplatzreservierung 198
4.4.1 MK_SPS_SPEICHERGROESSE 198
4.4.2 MK_SPS_DATENGROESSE 198
4.4.3 MK_SPV_SPEICHERGROESSE 198
4.4.4 MK_SPV_SYMBOLANZAHL 199
4.4.5 MK_HEADERANZAHL 199
4.4.6 MK_KANALANZAHL 199
4.4.7 MK_LAH_VORLAUFTIEFE 200
4.4.8 MK_LAH_RUECKLAUFGRENZE 200
4.4.9 MK_PFELD_GROESSE 201
4.5 Konfiguration der Achsen - Grundlegendes 202
4.5.1 MK_CANDRIVES 202
4.5.2 MK_APPLACHSIDX 203
4.6 Konfiguration der Achsen - Zuordnung und Auswertung 205
4.6.1 MK_CANDRIVES 205
4.6.2 MK_APPLACHSIDX 205
4.6.3 MK_ACHSENART 206

4.7 Konfiguration der Achsen - Auflösung 207
4.7.1 MK_IMPULSE 207
4.7.2 MK_WEG 207
4.7.3 MK_MASSSTAB 207
4.8 Konfiguration der Achsen - Arbeitsfeld 208
4.8.1 MK_GRUNDOFFSET 208
4.8.2 MK_SW_ENDS_MINUS, MK_SW_ENDS_PLUS 208
4.8.3 MK_SW_ENDS_MIT_RAMPE 208
4.8.4 MK_SCHLEPPGENAUHALT 208
4.8.5 MK_GENAUHALTZEIT 208
4.9 Konfiguration der Achsen - Reglereinstellung 209
4.9.1 MK_T2 209
4.10 Konfiguration der Achsen - Referenzierung 210
4.10.1 MK_REF_RICHTUNG_UND_FOLGE 210
4.11 Konfiguration der Achsen - Geschwindigkeit und Beschleunigung 211
4.11.1 MK_MODVMAX 211
4.11.2 MK_VMAX 211
4.11.3 MK_BESCHL, MK_BREMS 211
4.11.4 MK_T_BESCHL 211
4.11.5 MK_VBAHNMAX 212
4.11.6 MK_BAHNBESCHL, MK_BAHNBREMS 212
4.11.7 MK_T_BAHNBESCHL 212
4.12 Konfiguration der Achsen - Achskorrektur 213
4.12.1 MK_SPINDELUMKEHRSPIEL 213
4.13 Konfiguration der Achsen - Handräder 215
4.13.1 MK_CANDRIVES 215
4.13.2 MK_APPLACHSIDX 215
4.13.3 MK_ACHSENART 215
4.13.4 MK_IMPULSE 215
4.13.5 MK_WEG 215
4.13.6 MK_HANDRADZUORDNUNG 216
4.13.7 MK_HANDRADFAKTOR 216
4.13.8 MK_HANDRADFILTER 216
4.14 Konfiguration der Achsen - Synchronachsen 217
4.14.1 MK_ACHSENART 217
4.14.2 MK_SYNCHRONABWEICHUNG 217
4.14.3 MK_SYNCHRONOFFSET 217

4.15 Technologiespezifische Einstellungen 218
4.15.1 MK_MFKT_UPR_TABELLE 218
4.15.2 MK_TECHNOLOGIEDATEN1 ... MK_TECHNOLOGIEDATEN4 . . . 218
4.15.3 MK_MASCH_POLAR_KART 219
4.15.4 MK_KARTESISCH_ACHSNR 219
4.15.5 MK_POLAR_ACHSNR 220
4.15.6 MK_WLK_C_GRENZWINKEL 220
4.15.7 MK_WLK_C_OFFSET 220
4.15.8 MK_WLK_VERWEILZEIT 221
4.15.9 MK_X_WINKEL 221
4.15.10 MK_GEWINDE_VMAX 221
4.15.11 MK_DW224_255 222
4.16 Liste der Maschinenkonstanten 223

5 Schnittstelle SPS <-> NC-Betriebssystem 237

5.1 Definitionen 237
5.1.1 Datenbaustein 0 239
5.1.2 Datenbaustein 1 242
5.1.3 Datenbaustein 2 262
5.2 Erweiterte Schnittstelle für MMI-Funktionen 264
5.2.1 Datenbausteine 8 ... 14 264
5.2.2 Datenbaustein 15 267

6 ETC-MMI-Gateway 270

6.1 ETC-MMI-Gateway installieren 270
6.2 ETC-MMI-Gateway starten 271
6.3 ETC-MMI-Gateway konfigurieren 272
6.3.1 Connection - Verbindungen einrichten 272
6.3.2 Trace – Fehlerlogbuch 275
6.3.3 About – Versionsinformationen 276
6.4 Mmigtway.ini 277
6.4.1 Beispiel der Datei ”mmigtway.ini” 278
6.5 Kommunikationswerte im DPR Bereich 279

7 ETC-MMI 280
7.1 ETC-MMI installieren 280
7.2 ETC-MMI starten 282
7.3 ETC-MMI bedienen 283
7.3.1 Anzeigeelemente der Programmoberfläche 283
7.3.2 Bedienelemente der Programmoberfläche 284
7.3.3 Hilfefunktion 285
7.3.4 Konfigurationsdatei 285
7.3.5 Sprachumschaltung 285
7.3.6 Passworte 286
7.3.7 Hinweise, Warnungen, Fehlermeldungen 286
7.4 Betriebsart ”Einrichten” 287
7.5 Betriebsart ”Automatik” 292
7.6 Betriebsart ”Programmieren” 296
7.6.1 ASCII-Editor 299
7.6.2 Dateimanager 302
7.6.3 Zyklenprogrammierung 303
7.7 Betriebsart ”Diagnose” 306
7.8 Anhang 314
7.8.1 Sprachdatei (SPRACHE.TXT) 314
7.8.2 Zyklenprogrammierung 318
7.8.3 Konfigurationsdatei (DELPHMMI.INI) 320

8 SPS-Programmierung 329
8.1 ETC-SPS-Programmierung mit CoDeSys 329
8.2 CoDeSys installieren 330
8.2.1 Systemvoraussetzungen für CoDeSys V2.xx 330
8.2.2 Software installieren 330
8.3 ETC und PC verbinden 331
8.3.1 V.24-Schnittstelle 331
8.3.2 Ethernet-Schnittstelle (nur ETCHx, Hutschienenausführung) . 331
8.3.3 DPR-Schnittstelle (nur ETCPx, PCI-Steckkarte) 332
8.4 Projektierung 333
8.4.1 Zielsystem einstellen 333
8.4.2 SPS-Tasks der ETCxM konfigurieren 333
8.4.3 SPS-Tasks der ETCxC konfigurieren 335
8.4.4 E/A-Module konfigurieren 337
8.4.5 Adressierung 342
8.4.6 Remanente Variablen 343
8.4.7 Objektverzeichnis (Parametermanager) 343
8.4.8 Einstellungen in den globalen Variablenlisten im Objektverzeichnis 345

8.5 Netzwerkvariablen 346
8.5.1 Einstellungen in den Zielsystemeinstellungen 346
8.5.2 Einstellungen in den globalen Variablenlisten 347
8.6 Programm generieren 349
8.7 Schnittstelle zur ETC 350
8.7.1 Datenbausteine 350
8.7.2 Systemvariablen der ETCxC 350
8.7.3 Systemvariablen der ETCxM 353
8.7.4 Maschinenkonstanten in der ETCxC verwenden 353
8.7.5 Maschinenkonstanten in der ETCxM verwenden 353
8.7.6 Parameterfeld der ETCxC 353
8.7.7 Betriebsdaten der ETCxC 354
8.7.8 Fehlermeldungen bei der ETCxM auslesen 354
8.8 Bibliothek 355
8.8.1 Allgemeine Funktionen 355
8.8.2 V24-Funktionen 370
8.8.3 FILE IO-Funktionen 375
8.8.4 Speicherzugriffsfunktionen 384
8.8.5 CANopen Funktionen 389
8.8.6 CAN-Funktionen (nur ETCxM) 400
8.8.7 MMI-Kommunikationsfunktionen 405
8.8.8 Echtzeituhr (nur ETCxM) 408
8.9 Bibliothek ServerSDO.lib 409
8.9.1 InitServerSdo 409

9 Stichwortverzeichnis 410

1 Vorwort und Allgemeines

1.1 Über dieses Handbuch

Zielgruppe

Dieses Handbuch wendet sich an alle Personen, die das ETC Motion Control-System unter dem Betriebssystem ”NC” programmieren und inbetriebnehmen.

Inhalt

Im Softwarehandbuch ”Betriebssystem NC” finden Sie Informationen zu folgenden Themen:

- Erste Schritte – eine chronologische Beschreibung der Inbetriebnahmeschritte
- CNC-Programmierung – Arbeiten mit G-Funktionen, Formelprozessor, Satzerweiterungen und Datenfeldern
- Maschinenkonstanten – Antriebe parametrieren
- SPS-ETC-Schnittstelle – Datenbausteine und ihre Funktionen
- ETC-MMI-Gateway – zwischen Windows-Applikationen und der ETC kommunizieren
- ETC-MMI – Entwicklungsumgebung für die Erstellung von CNC-Programmen
- CoDeSys – Entwicklungsumgebung für die Erstellung von SPS-Programmen

Weitere Information

Im hardware-orientierten ETC-Gerätehandbuch finden Sie Informationen zu folgenden Themen:

- Technische Daten
- Aufbau und Funktion der System-Komponenten einschließlich Schnittstellen-Beschreibung
- System-Komponenten montieren, anschließen und instandsetzen

1.2 Für welche Produkte ist das Handbuch gültig?

Grundgerät

①					
ETC	xx	0	xx	1A	10

Produkt

Ausführung
HC = Hutschiene, CNC-Kern
PC = PCI-Steckkarte, CNC-Kern

Anzahl Achsen
02 = 2 Achsen
04 = 4 Achsen
08 = 8 Achsen
12 = 12 Achsen

Hardwarestand

Softwarestand

ETCHC0xx

ETCPC0xx

Module

①			
ETCH	xxxx	1A	10

Produkt

N003 = Netzteil
T000 = Busabschlussmodul
I008 = 8 dig. Eingänge
I016 = 16 dig. Eingänge
U008 = 8 dig. Ausgänge
U016 = 16 dig. Ausgänge
A022 = je 2 analoge Ein- und Ausgänge

Hardwarestand

Softwarestand

ETCHxxxx

1.3 Rechtliche Bestimmungen

Kennzeichnung

Die Komponenten des ETC Motion Control-Systems sind eindeutig durch den Inhalt des Typenschilds gekennzeichnet.

Hersteller

Lenze Drive Systems GmbH, Postfach 101352, D-31763 Hameln

CE-Konformität

Konform zur EG-Richtlinie "Elektromagnetische Verträglichkeit"

Bestimungsgemäße Verwendung

Komponenten des ETC Motion Control-Systems

- ► nur unter den im ETC-Gerätehandbuch vorgeschriebenen Einsatzbedingungen betreiben.
- ► sind nicht zugelassen für den Einsatz in explosionsgefährdeten Umgebungen.
- ► erfüllen die Schutzanforderungen der EG-Richtlinie "Niederspannung".
- ► sind keine Maschinen im Sinne der EG-Richtlinie "Maschinen".
- ► sind keine Haushaltsgeräte, sondern als Komponenten ausschließlich für die Weiterverwendung zur gewerblichen Nutzung bestimmt.

Die Verantwortung für die Einhaltung der EG-Richtlinien in der Maschinenanwendung liegt beim Weiterverwender.

Jede andere Verwendung gilt als sachwidrig!

Haftung

Die in diesem Handbuch angegebenen Informationen, Daten und Hinweise waren zum Zeitpunkt der Drucklegung auf dem neuesten Stand. Aus den Angaben, Abbildungen und Beschreibungen in diesem Handbuch können keine Ansprüche auf Änderung bereits gelieferter Systeme und -Komponenten geltend gemacht werden.

Die in diesem Handbuch dargestellten verfahrenstechnischen Hinweise und Schaltungsausschnitte sind Vorschläge, deren Übertragbarkeit auf die jeweilige Anwendung überprüft werden muss. Für die Eignung der angegebenen Verfahren und Schaltungsvorschläge übernimmt Lenze keine Gewähr.

Die Angaben in diesem Handbuch beschreiben die Eigenschaften der Produkte, ohne diese zuzusichern. Es wird keine Haftung übernommen für Schäden und Betriebsstörungen, die entstehen durch:

- ► Missachten des Handbuchs
- ► Eigenmächtige Veränderungen an Komponenten des ETC-Motion Control-Systems
- ► Bedienungsfehler und unsachgemäßes Arbeiten an und mit dem ETC Motion Control-Systems

Gewährleistung

Siehe Verkaufs- und Lieferbedingungen der Lenze Drive Systems GmbH.

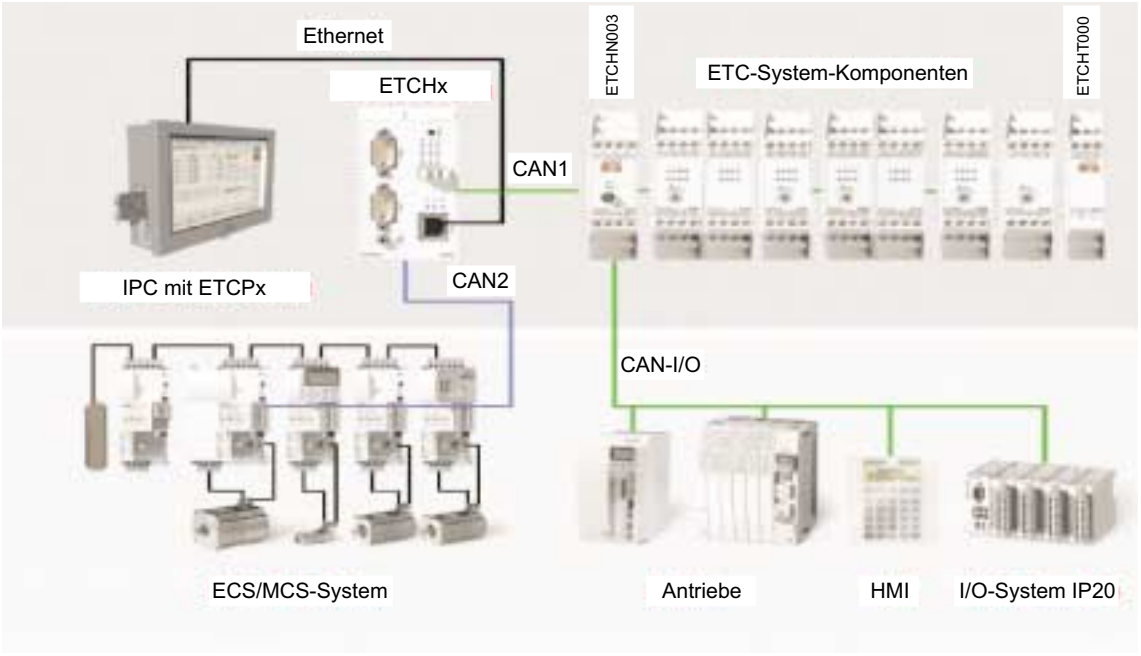
Gewährleistungsansprüche sofort nach Feststellen des Mangels oder Fehlers bei Lenze anmelden. Die Gewährleistung erlischt in allen Fällen, in denen auch keine Haftungsansprüche geltend gemacht werden können.

2 Erste Schritte

Dieses Kapitel vermittelt Grundlagen des ETC-Systems und beschreibt die Vorgehensweise bei der Realisierung einer Automatisierungsaufgabe.

2.1 Systemübersicht

2.1.1 Beispiel für ein Automatisierungssystem



ETCHx / ETCPx

Kern des Automatisierungssystems ist die ETC-Steuerung in Hutschienen-Ausführung (ETCHx) oder als PCI-Steckkarte (ETCPx).

Die Hutschinenen-Ausführung ETCHx kommt in der Regel in einer sogenannten ETC-Insel zum Einsatz (📖 15); sie kommuniziert über Ethernet mit den Applikationen auf dem IPC (oder Standard-PC).

Die ETCPC steckt im IPC (oder Standard-PC) und kommuniziert über den internen PCI-Bus mit den Applikationen.

Beide Ausführungen besitzen zwei getrennte CAN-Busse:

- Am CAN1-Bus (oder auch ME-Bus) werden E/A-Module und evtl. Bedienteile angeschlossen. Fremdgeräte müssen das Profil DS401 der CANopen-Spezifikation erfüllen.
- An den auf die Frontplatte geführten Anschluss für den Motion-CAN-Bus (CAN2) werden die Antriebe angeschlossen (z. B. der ECS-Kompaktservo). Fremdantriebe müssen das Profil DS402 der CANopen-Spezifikation erfüllen und speziell den ”Interpolated Position Mode” unterstützen.

Damit die ETC-Steuerung ihre zugedachte Steuerungsaufgabe ausführen kann, benötigt sie verschiedene Programme, die vom IPC (oder Standard-PC) übertragen werden:

- Betriebssystem bzw. Firmware der Steuerung (z. B. ETCHC.rsc)
- SPS-Programme (z. B. SPSDummy.prg)
- CNC-Programme; d. h. Zyklen- und DIN-Programme (z. B. 9000.zyk oder Nikolaus.din)

IPC

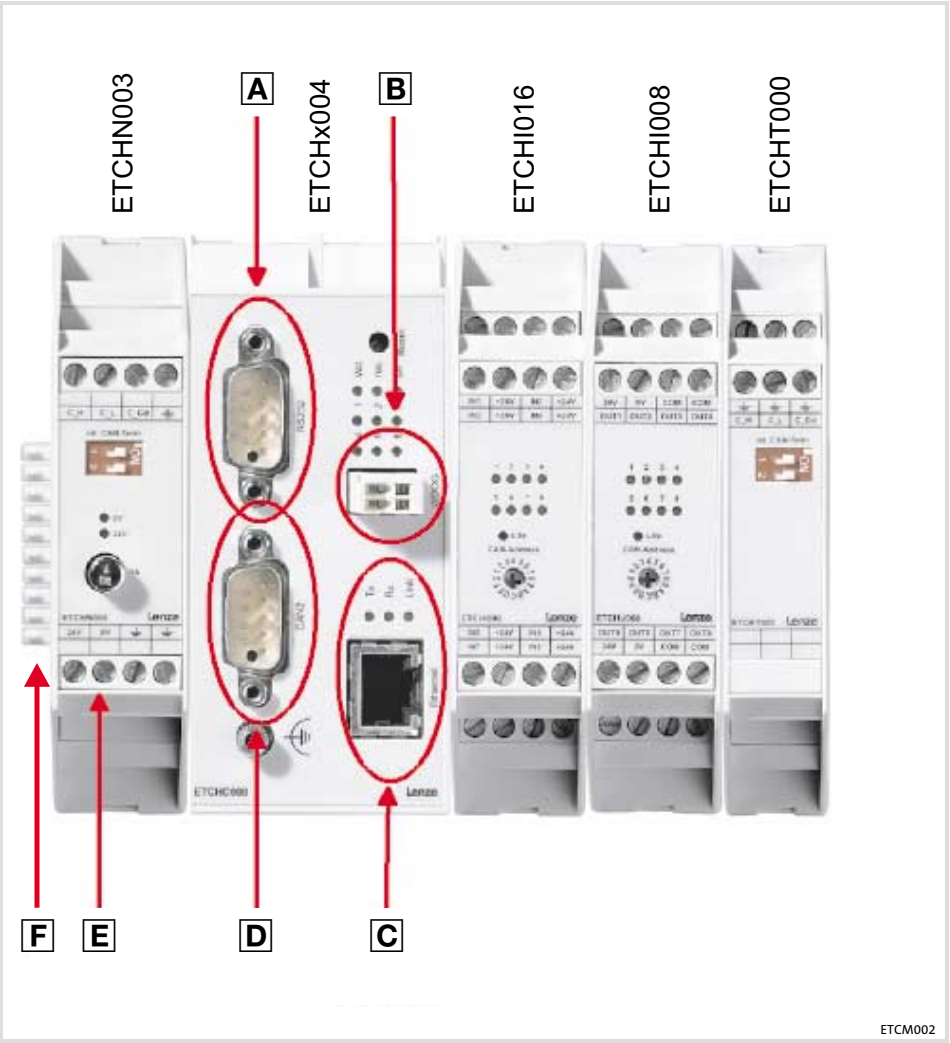
Über den IPC (oder Standard-PC) programmieren, bedienen und warten Sie die ETC-Steuerung. Hier laufen z. B. folgende Applikationen:

- Betriebssystem Windows 2000 oder XP
- Terminal-Programm (z. B. HyperTerminal) zum Einrichten der Ethernet-Verbindung zwischen IPC und ETCHx-Steuerung und für die Konfiguration der ETCHx-Steuerung über die Monitorschnittstelle (z. B. Update der Firmware). Für die ETCPx ist es nicht erforderlich.
- ETC-MMI-Gateway als Kommunikationsprogramm zwischen Windows-Applikationen und ETC-Steuerung.
- Lenze-ETC-MMI zum Konfigurieren, Bedienen und Beobachten der ETC-Steuerung und zum Erstellen von CNC-Programmen.
- CoDeSys-Entwicklungsumgebung zum Erstellen und Testen von SPS-Programmen.

Hinweis!

Für die Inbetriebnahme benötigen Sie eine externe Tastatur am IPC. Während des normalen Betriebs ist sie nicht erforderlich.

2.1.2 Beispiel für den Aufbau einer ETC-Insel



A Serielle Schnittstelle (RS232)
B Watchdog (z. B. für Not-Aus-Kette)
C Ethernet
D Motion-CAN-Bus (CAN2)
E 24 V-Versorgung
F ME-Bus (CAN1)

ETCHN003: Netzteil zur Versorung der ETC-Insel und ME-Busabschluss

ETCHx004: ETC Motion Control für 4 Achsen (Steuerung)

ETCHIxxx: Eingangsmodule mit 16 bzw. 8 digitalen Eingängen

ETCHT000: ME-Busabschlussmodul

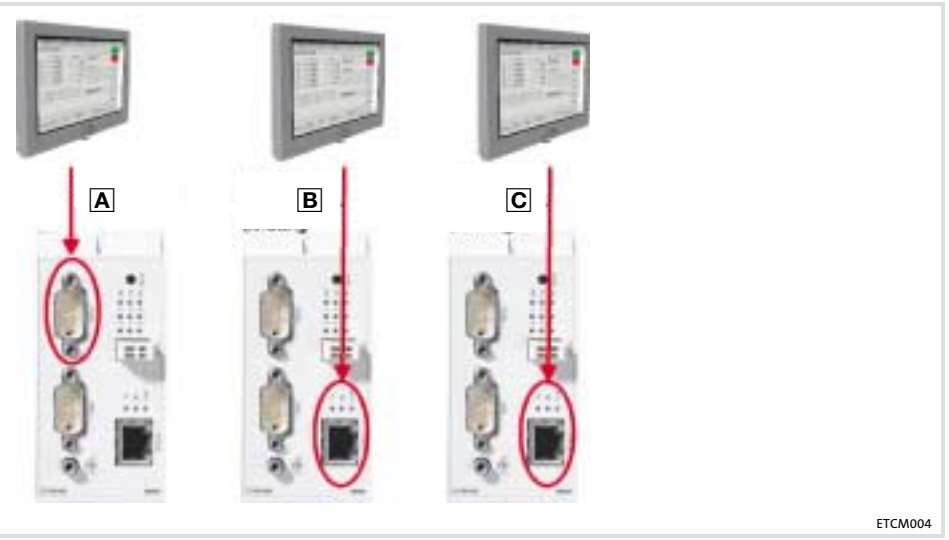
Hinweis!

Für den Abschluss des ME-Busses ist sowohl beim Netzteil ETCHN003 als auch beim Busabschlussmodul ETCHT000 der DIP-Schalter 1 auf ON zu stellen.

2.1.3 ETCHx und PC verbinden

Die Verbindung zwischen der ETCHx und einem PC ist auf 3 Arten möglich.

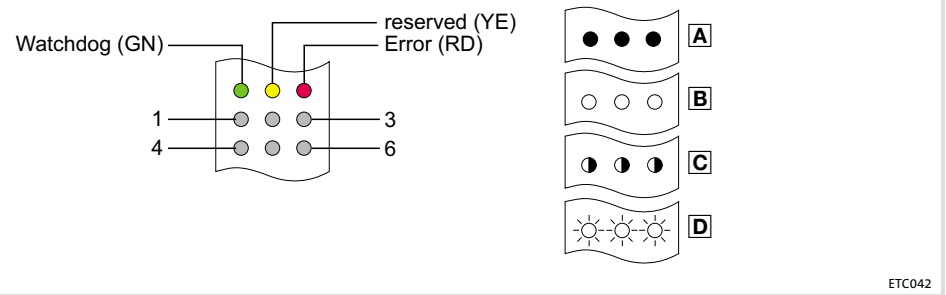
Verbindungsart	Leitungsausführung	Beschreibung
Serielle Verbindung	Systemleitung Typ EWL 0068 oder vergleichbarer RS232-Leitung mit beidseitigr 9-poliger SUB-D-Buchse (Pin-Belegung siehe ETC-Gerätehandbuch)	Diese Verbindung wird nur für die Inbetriebnahme benötigt! Ein freier COM-Port am PC wird mit der RS232-Schnittstelle der ETCHx verbunden. Der PC kommuniziert über ein Terminalprogramm mit der Monitorschnittstelle der ETCHx.
Lokale Verbindung über Ethernet (Pear-to-Pear-Verbindung)	Systemleitung Typ EWL 0065, EWL 0066 oder EWL 0067 (beidseitig RJ45)	Eine Netzwerkkarte im PC wird mit dem Ethernet-Anschluss der ETCHx über ein Cross-over-Kabel verbunden. Kommuniziert wird über das TCP/IP-Protokoll.
Netzwerk-Verbindung	Ethernet Patch-Kabel STP Cat5 (beidseitig RJ45)	Die ETCHx wird über ihren Ethernet-Anschluss an ein separates oder bestehendes Netzwerk (z. B. Intranet) angeschlossen. Kommuniziert wird über das TCP/IP-Protokoll.



A Verbindung über serielle Schnittstelle
B Lokale TCP/IP-Verbindung
C Netzwerk TCP/IP-Verbindung

2.2 Statusanzeige

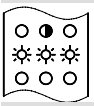
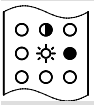
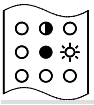
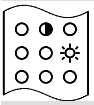
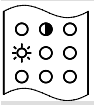
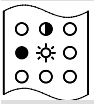
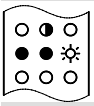
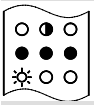
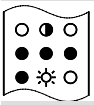
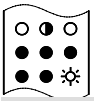
LEDs auf der Frontplatte der ETC melden den aktuellen Status des Systems. Die Bedeutung der Signale unterscheiden sich in der Hochlaufphase und im Betrieb.



A LED ein
B LED aus
C LED beliebig
D LED blinkt

Hochlaufphase

Zu Beginn des Hochlaufs wird ein RAM-Test durchgeführt. Nach dem fehlerfreien RAM-Test erzeugen die LEDs 1 ... 6 ein Lauflicht. Etwaige Fehler während des Bootups werden durch die nachfolgenden Blink- und Anzeigemuster signalisiert.

	Checksummenfehler im internen FLASH-PROM. Blinkt 3-mal schnell hintereinander. Danach wird der Bootloader neu ins interne FLASH-PROM gebrannt. Tritt immer nach einem Update des Bootloaders auf.
	Fehler in den letzten 32 kByte des RAM-Speichers (nach einem Update und Schreiben des Bootloaders in das Flash).
	Fehler in den ersten 32 kByte des RAM-Speichers.
	Fehler beim Entpacken des Bootloaders.
	Fehler beim Brennen des Bootloaders ins FLASH-PROM.
	Fehler im RAM-Speicher.
	Fehler in den letzten 32 kByte des RAM-Speichers.
	Keine Firmware geladen oder Checksummenfehler in der Firmware. Nach 5-mal Blinken wird der Bootmonitor aktiviert.
	Fehler beim Laden der Firmware aus dem FLASH-PROM. Nach 5-mal Blinken wird der Bootmonitor aktiviert.
	Fehler beim Starten der Firmware.

Betrieb

Wenn die Steuerung nach der Hochlaufphase in den Betriebszustand geht, gelten die nachfolgenden Blink- und Anzeigemuster.

LED	Bedeutung
WATCHDOG	Watchdog, muss im hochgelaufenen Zustand immer leuchten.
reserviert	Ohne Funktion
ERROR	Blinkt nach Exception (Schutzverletzung des Steuerungsprogramms während der Laufzeit, Verlassen des erlaubten Variablenbereichs, Division durch Null etc.).
LED 1	Blinkt im Takt des Grobinterpolators
LED 2	Blinkt im Takt des Interpreters
LED 3	Blinkt im Takt der Zentralsteuerung
LED 4	Blinkt im Takt des Feininterpolators
LED 5	Blinkt bei jedem RS232-Interrupt oder CAN-Interrupt
LED 6	Blinkt im Takt der SPS-Zykluszeit

Hinweis!

Die ”Blink-Frequenz” kann bei kurzen Programmen so niedrig sein, dass die LED’s dunkel erscheinen.

2.3 Inbetriebnahmeschritte (Übersicht)

Stop!

Beachten Sie die Hinweise im Kapitel ”Erstes Einschalten” des ETC-Gerätehandbuchs bevor Sie das System in Betrieb nehmen.

Hinweis!

Die PCI-Steuerungsvariante ETCPx erst in Schritt 6 einbauen und installieren.

Schritt	ETCHC	ETCPC	Beschreibung	Siehe
1	X	-	ETCHC über RS232-Leitung mit PC verbinden und starten.	🕮 21
2	X	-	Am PC Terminalprogramm starten (ggf. vorher konfigurieren) und Monitorschnittstelle der ETCHC aktivieren.	🕮 21
3	X	-	Wenn die ETCHC in der Betriebsart ”mit MMI” betrieben werden soll, auf der ETC die Firmware-Datei durch die Datei ”NetBoot.rsc” ersetzen.	🕮 25
4	X	-	IP-Adresse der ETCHC zuweisen.	🕮 28
5	X	X	Ggf. ETC-MMI und ETC-MMI-Gateway installieren.	🕮 31
6	-	X	ETCPC in PC einbauen und Treiber installieren.	🕮 31
7	X	X	ETC-MMI und ETC-MMI-Gateway starten.	🕮 32
8	X	X	Verbindung zwischen ETC-MMI und ETCxC herstellen.	🕮 34
9	X	X	Antriebe über Maschinenkonstanten parametrieren.	🕮 36
10	X	X	Parameter der Antriebe überprüfen.	🕮 43
11	X	X	Antriebe im Tippbetrieb testen.	🕮 44
12	X	X	CNC-Programm im ETC-MMI erstellen.	🕮 47
13	X	X	CNC-Programm in die ETCxC laden, starten und testen.	🕮 49
14	X	X	Ggf. CoDeSys installieren.	🕮 51
15	X	X	CoDeSys starten und konfigurieren.	🕮 51
16	X	X	SPS-Programm erstellen.	🕮 58
17	X	X	SPS-Programm in die ETCxC laden, starten und testen.	🕮 66
18	X	X	CNC- und SPS-Programm testen.	🕮 70

2.4 Kommunikation zwischen PC und ETCHx herstellen

Hinweis!

Die in diesem Kapitel beschriebenen Schritte gelten nur für die ETCHx-Variante (Hutschienen-Variante); für die ETCPx-Variante (PCI-Karte) sind sie nicht erforderlich.

2.4.1 ETCHx starten

1. Verbinden Sie die seriellen Schnittstellen von PC und ETCHx.
   Verwenden Sie dazu die Systemleitung Typ EWL 0068 oder eine vergleichbare RS232-Leitung mit beidseitiger 9-poligen D-Sub-Buchse (Pinbelegung siehe ETC-Gerätehandbuch).
2. Starten Sie den PC.
3. Starten Sie die ETCHx durch Netzeinschalten oder initialisieren Sie die ETCHx durch ein Reset (Reset-Taste auf Frontplatte drücken).
   Sobald die ETC-Firmware geladen ist, leuchtet an der ETCHx die grüne Watchdog-LED.

2.4.2 Terminalprogramm ”HyperTerminal” starten und Monitorschnittstelle aktivieren

Für die Kommunikation zwischen PC und ETCHx benötigen Sie ein Terminal-Programm. Das Terminalprogramm ”HyperTerminal” steht in jeder Standard-Windows-Installation zur Verfügung.

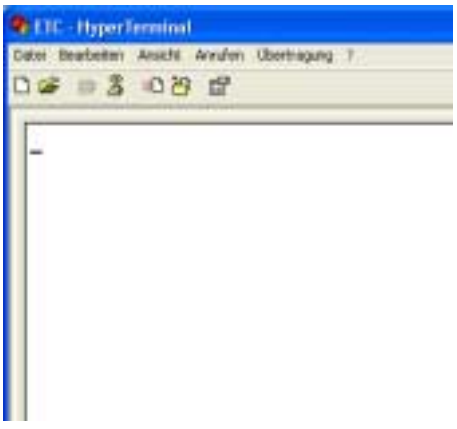
1. Falls Sie bereits eine Verbindung mit einer ETC hergestellt hatten (d. h. falls bereits eine Verbindung konfiguriert ist), starten Sie die Verbindung mit **<Start> → Programme → Zubehör → Kommunikation → HyperTerminal →** *Verbindungsname.ht***.** Weiter im nächsten Abschnitt.
   Falls noch keine Verbindung konfiguriert ist, starten Sie das HyperTerminal über **<Start> → Programme → Zubehör → Kommunikation → HyperTerminal**.
   Das Programm fragt verschiedene Einstellungen des Telefonanschlusses ab, da es auch für eine Modem-Verbindung gedacht ist. Diese Einstellungen sind hier nicht von Bedeutung.
2. Nach Aufforderung vergeben Sie einen Namen für die Verbindung (zum Beispiel ”ETC”) und klicken auf **OK**.
3. Wählen Sie im Dialog *”Verbinden mit”* die PC-Schnittstelle, über die Sie die Verbindung herstellen möchten (zum Beispiel ”COM1”).

ETCM006

4. Klicken Sie auf **OK**.
5. Geben Sie im Dialog ”Eigenschaften von COMx” folgende Daten ein:

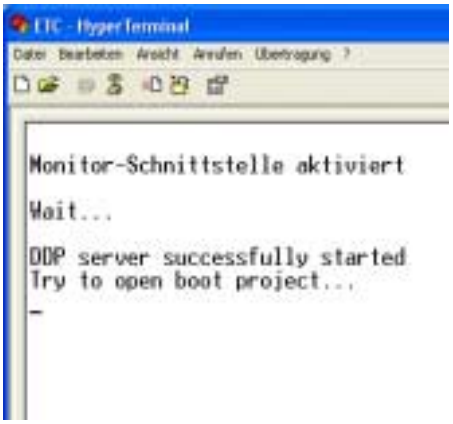
ETCM007

6. Klicken Sie auf **OK**.



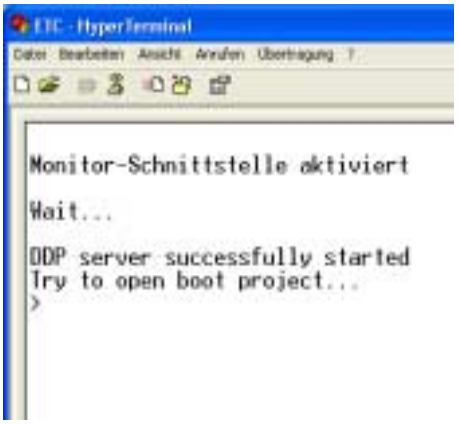
ETCM008

Sobald eine Verbindung zwischen PC und ETC besteht, blinken an der ETCHx die LEDs 1 ... 6 umlaufend. Die Meldung ”Wait Boot Loader” erscheint und anschließend folgendes Fenster:



ETCM009

7. Drücken Sie die Taste **<?>** bis der Prompt ">" erscheint.



ETCM010

Die Monitorschnittstelle ist gestartet. Sie können jetzt über die Tastatur Befehle zur ETC senden.

Wichtige Befehle des Monitorprogramms

Befehl	Bedeutung
dir [dr:][pattern]	Zeigt das Inhaltsverzeichnis des angegebenen Laufwerks an. Voreingestellt ist die Flashdisk (sd:), andere mögliche Laufwerke sind Programmspeicher (ps:), Ram-Disk (rd:) und Floppy-Disk (fd:), falls vorhanden. Als Pattern sind die gängigen MS-DOS-Pattern verwendbar, z. B.: *.din
del [dr:][pattern]	Löscht die angegebenen Dateien auf dem angegebenen Laufwerk. Voreingestellt ist auch hier die Flash-Disk (sd:).
cd [dr:]	Wechselt das aktuell ausgewählte Laufwerk bzw. zeigt es an. Voreingestellt ist die Flash-Disk (sd:).
format dr:	Formatiert das aktuelle Laufwerk und löscht damit alle darauf vorhandenen Dateien. Ein Laufwerkname muss angegeben werden.
reboot	Reset der Steuerung. Um den Bootmonitor zu aktivieren, muss anschließend die Taste "!" gedrückt werden, bis sich der Bootmonitor meldet (kann 3-4 Sekunden dauern).
help	Hilfeseite mit den Befehlen des Monitorprogramms anzeigen.
ver	Versionstring anzeigen
ipconf	Netzwerkparameter für Ethernetschnittstelle konfigurieren
hwconf	Hardware-Konfiguration der Steuerung anzeigen (CPU-Typ, Speicher usw.)

Störungsbeseitigung

Falls mit der beschriebenen Vorgehensweise keine Verbindung zur ETCHx aufgebaut wird, gehen Sie wie folgt vor:

1. Kontrollieren Sie die Verkabelung zwischen PC und ETCHx.
2. Kontrollieren Sie die Verbindungsparameter des Terminalprogramms.

Ein erster Test der Kabelverbindung lässt sich mittels einer Brücke zwischen den Pins 2 und 3 an der Buchse des Kabels (Steuerungsseite) durchführen:

► Das Terminalprogramm wird danach wie bisher aufgerufen und beliebige Buchstaben über die Tastatur eingegeben.

► Falls die eingegebenen Zeichen der Tastatur in der Maske des Terminalprogramms korrekt angezeigt werden (die Zeichen werden durch die Brücke als Echo zurückgesandt), ist das Kabel auf eine vertauschte Pinbelegung (Pin2/3), auf Kurzschluss und die Verbindung auf richtige Parametrierung zu kontrollieren.

2.4.3 Betriebsart der ETCHx einstellen

Die ETCHx kann in zwei Betriebsarten betrieben werden:

- Variante ”Stand-Alone” (Auslieferungsvariante)
  - Die Steuerung lädt direkt nach Anlegen der Spannung die Firmware.
  - Die Steuerung führt ein festes Programm aus.
- Variante ”mit MMI”
  - Die Steuerung wartet nach Anlegen der Spannung auf das Laden einer Firmware von einem PC über ein beliebiges MMI-Programm (z. B. ETC-MMI).
  - Variable Anwenderprogramme können ausgeführt werden.

Vergleich der Betriebsarten

	Stand-Alone (Auslieferung)	Mit MMI
Einsatz	Die Steuerung betreibt eine Maschine ohne weitere Kopplung zu einem PC.	Die Steuerung betreibt eine Maschine mit Kopplung zu einem PC mit Bedienoberfläche, der zum Laden von Programmen und zum Bedienen der Steuerung dient.
Anwenderprogramm	In der Steuerung läuft ein festes Programm.	Variable Anwenderprogramme können in die Steuerung geladen werden.
Installierte Firmware auf der Steuerung	Bootloader Firmware	Bootloader Loader
Installation der Firmware	Die Firmware wird über den Bootloader in das EEPROM der Steuerung geladen und installiert.	Der Loader wird über den Bootloader in das EEPROM der Steuerung geladen und installiert.
Verhalten der Steuerung nach Anlegen der Spannung	Die Steuerung startet nach Anlegen der Spannung automatisch mit ihrer Firmware.	Die Steuerung wartet nach Anlegen der Spannung auf das Laden der Firmware über den PC.
LED Anzeige nach Anlegen der Spannung	Nach erfolgreichem Start der Firmware leuchtet die grüne Watchdog-LED.	Der erfolgreiche Start des Loaders wird durch umlaufende LEDs (1 ... 6) angezeigt.
Starten der Firmware	Die Firmware wird vor dem Start aus dem EEPROM der Steuerung in das RAM geladen und gestartet.	Die Firmware befindet sich auf dem PC und wird beim Start der MMI-Bedienoberfläche in das RAM der Steuerung geladen und gestartet. Nach erfolgreichem Start der Firmware leuchtet die grüne Watchdog-LED.

Betriebsart ”Stand-Alone” einstellen

Die Betriebsart ”Stand-Alone” ist werkseitig eingestellt.

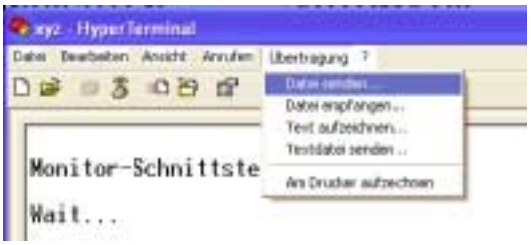
Hinweis!

Falls die ETCHx schon mal in der Betriebsart ”mit MMI” betrieben wurde, dann aber wieder in die Betriebsart ”Standalone” zurückgesetzt werden soll, muss in der Steuerung die Datei ”NetBoot.rsc” (Loader) gelöscht und stattdessen die Firmware-Datei ETCHC_A.rsc (ETCHC) oder ETCPC.rsc (ETCPC) übertragen werden. Gehen Sie dabei wie im folgenden Abschnitt beschrieben vor; der Unterschied besteht nur in den Dateien, die ausgetauscht werden.

Betriebsart ”Mit MMI” einstellen

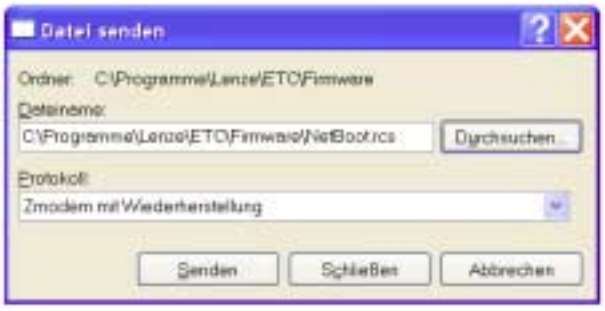
Für die Betriebsart ”mit MMI” muss auf der ETCHx die Firmware-Datei durch die Datei ”NetBoot.rsc” (Loader) ersetzt werden.

1. Um die Firmware zu löschen, geben Sie folgendes im HyperTerminal ein: **del sd: etc*.rsc**. Drücken Sie abschließend **<Enter>**.
2. Um den Loader auf die ETC zu übertragen, aktivieren Sie im HyperTerminal den Menüpunkt **Übertragung → Datei senden**.



ETCM021

3. Geben Sie den Pfad zur Datei ”NetBoot.rsc” vor und wählen Sie als Protokoll ”Zmodem mit Wiederherstellung”.



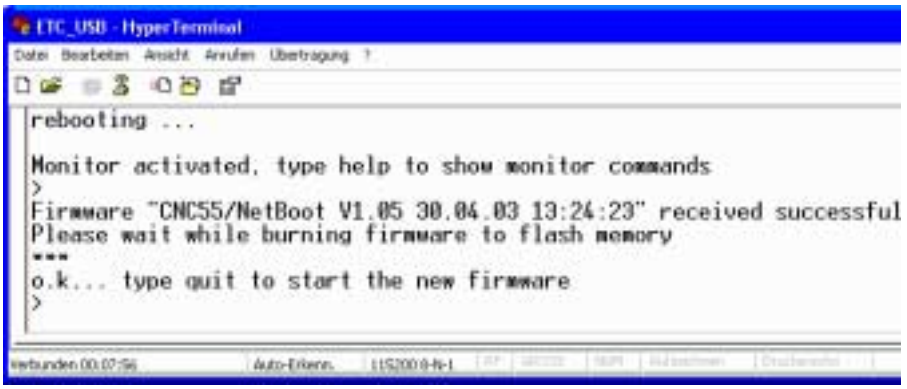
ETCM022

4. Klicken Sie auf **Senden**.

   Während des Ladens der Datei in die Steuerung erscheint das nachfolgende Bild. Auf der ETC-Frontplatte blinken die LEDs 1 ... 6 umlaufend.

ETCM023

Nach Abschluss des Ladevorgangs erscheint das folgende Bild:


ETC_USB - HyperTerminal
Datei Bearbeiten Ansicht Anrufen Übertragung ?
rebooting ...
Monitor activated, type help to show monitor commands
>
Firmware "CNC55/NetBoot V1.05 30.04.03 13:24:23" received successful
Please wait while burning firmware to flash memory
***
o.k... type quit to start the new firmware
>

ETCN007

Die Steuerung wartet auf das Laden der Firmware vom PC; auf der Frontplatte der ETC leuchten die LEDs 1 ... 6 umlaufend.

5. Geben Sie imHyperTerminal-Fenster **quit** ein und bestätigen Sie den Befehl mit **<Enter>** .

   Die Firmware startet. Auf der ETC-Frontplatte leuchtet die grüne Watchdog-LED.

2.4.4 IP-Adresse der ETCHx zuweisen

Für die Kommunikation über ein Netzwerk oder ein lokales Ethernet-Kabel benötigt die ETCHx eine eindeutige und zu den anderen Teilnehmern passende IP-Adresse (mit Subnet-Maske). Bei Auslieferung der Steuerung hat die Steuerung eine bestimmte, aber zufällige IP-Adresse.

Die IP-Adresse der ETCHx weisen Sie über die Monitorschnittstelle zu, wie nachfolgend beschrieben. Diese IP-Adresse wird bei der Erstellung von Verbindungen im ETC-MMI-Gateway abgefragt (📖 34).

Bilden Sie die IP-Adresse nach den folgenden allgemeinen Regeln für IP-Adressen:

Separates (lokales) Netzwerksegment

Sind Steuerung und PC mit einem lokalen Netzwerksegment oder durch eine direkte Ethernet-Verbindung verbunden (ohne Internet/Intranet-Anschluss), ordnen Sie den beiden Netzwerkstationen IP-Adressen der Klasse B oder Klasse C zu.

- ► Adressen der Klasse B liegen im Bereich von 128.0.0.0 bis 191.255.255.255. Die entsprechende Subnetz-Maske ist 255.255.0.0 (default).
- ► Adressen der Klasse C liegen im Bereich von 192.0.0.0 bis 223.255.255.255. Die entsprechende Subnetz-Maske ist 255.255.255.0.

Die Gateway-Adresse ist hier nicht relevant. Sie können eine freie IP-Adresse des Subnetzes eingeben (aber nicht 0.0.0.0).

Bespiel	IP-Adresse	Subnetz-Maske	Gateway-Adresse
Diese Adresse ist der ETCHC zuzuweisen und als Verbindungsparameter im ETC-MMI-Gateway einzugeben 📖 34:	181.16.1.10	255.255.0.0	181.16.1.1
Diese Adresse ist in der Netzwerkverbindung des PCs einzugeben (TCP/IP-Einstellungen):	181.16.1.11	255.255.0.0	181.16.1.1

Hinweis!

Verwenden Sie in IP-Adressen keine führenden Nullen, da die IP-Adresse sonst als Oktadezimalzahl interpretiert wird.

Existierendes Netzwerksegment

Sollen ETCHx und PC mit einem existierenden Netzwerksegment verbunden werden, müssen die Adressen und die Subnetz-Maske vom Netzwerkadministrator vergeben werden.

Hinweis!

Wenn der Ethernet-Adapter 100Base-T-Technologie unterstützt, muss für die ”Media Type”-Eigenschaft ”Auto Select” oder ”10Base-T” eingestellt sein.

MAC –Adresse

Die ETCHx wie auch jedes andere Gerät mit Ethernet-Controller erhält vom Hersteller eine unveränderbare und weltweit eindeutige physikalische Ethernet-Adresse, auch MAC-ID (Media Access Control Identity) genannt. Diese kann zur Adressierung auf Hardware-Ebene verwendet werden. Die Adresse besitzt eine feste Länge von 6 Bytes (48 Bit) und beinhaltet einen Adresstyp, eine Kennzeichnung des Herstellers und eine Seriennummer.

MAC-ID der ETC: 00:05:7e:xx:xx:xx (mit xx:xx:xx = Seriennummer).

IP-Adresse zuweisen

Die IP-Adresse der ETCHx stellen Sie über die Monitorschnittstelle ein.

1. Geben Sie im HyperTerminal den Befehl **ipconf** ein und bestätigen Sie den Befehl mit **<Enter>**.

   Die MAC-ID und die aktuelle Adresseinstellung der ETCHx werden angezeigt.

ETCM011

2. Drücken Sie die Taste **<Y>** und ändern Sie die Einstellungen. Folgen Sie dabei den Anweisungen der Monitiorschnittstelle.

   Wählen Sie eine IP-Adresse gemäß den am Anfang dieses Abschnitts genannten Regeln.

   Das Programm HyperTerminal kann jetzt beendet werden.

Hinweis!

Notieren Sie die IP-Adresse, da sie bei der Erstellung von Verbindungen im ETC-MMI-Gateway eingegeben werden muss (📖 34).

2.5 ETC-MMI und ETC-MMI-Gateway konfigurieren

Das Programm ”ETC-MMI” wird für folgende Aufgaben verwendet:

- ► Steuerung konfigurieren
- ► Steuerung bedienen und beobachten
- ► Steuerung warten und Fehler diagnostizieren

Das MMI-Gateway ist das Kommunikationsprogramm zwischen Windows Applikationen und ETC-Steuerungen. Unterschiedliche Applikationen, wie MMIs, Konfigurationstools oder OPC-Server können gleichzeitig ihre Verbindungen zu einer oder mehreren Steuerungen aufbauen.

Das Gateway implementiert alle nötigen Mechanismen für Zugriffsverwaltung, Fehlerhandling und Diagnose und unterstützt steuerungsspezifische Hardwaretreiber und Kommunikationsprotokolle. Mit Hilfe des Konfigurationstools können neue Verbindungen eingefügt bzw. vorhandene Verbindungen editiert werden.

Das ETC-MMI-Gateway wird bei der Installation des Lenze-ETC-MMIs mitinstalliert. Es kann aber auch als separate Applikation installiert werden und steht dann als Kommunikationskanal für andere Applikationen zur Verfügung.

Weitere Informationen entnehmen Sie den Kapiteln ”ETC-MMI-Gateway” (🕮 270) und ”ETC-MMI” (🕮 280).

2.6 ETC-MMI installieren

Stop!

Die PCI-Steuerungsvariante ETCPx erst nach der ETC-MMI-Installation und vor dem ersten Starten des ETC-MMIs installieren.

Hinweis!

Das ETC-MMI-Gateway wird bei der Installation des Lenze ETC-MMIs mitinstalliert

1. Starten Sie über den Windows-Explorer das Programm ”setup.exe” auf der ETC-MMI-Installations-CD.
2. Folgen Sie den Anweisungen des Installationsprogramms. Folgendes wird abgefragt:
   – Demo- oder Standard-Installation
   – Steuerungstyp (ETCHC oder ETCPC)
   – IP-Adresse der ETCHC (📖 28) bzw. Index der ETCPC (Indizes der PCI-Karten liegen im Bereich 0 ... 9 und werden vom Treiber automatisch zugewiesen. Die erste ETCPx hat den Index 0).

   In der Startleiste wird der Eintrag ”Lenze ETC-MMI” angelegt, über den die Applikation gestartet werden kann.
3. Bei einer ETCHC: Überprüfen Sie , dass TCP/IP-Protokolle in Netzwerkeigenschaften installiert sind (**Start → Einstellungen → Systemsteuerung → Netzwerk**).

   Bei einer ETCPC: Bauen Sie die ETCPC-Steckkarte in den PC ein. Beachten Sie dabei die Hinweise im ETC-Gerätehandbuch.

2.6.1 ETCPx einbauen und installieren

1. Wenn Sie als Steuerung eine ETCPx-Steckkarte einsetzen, bauen Sie diese jetzt in den PC ein. Beachten Sie dabei die Hinweise im ETC-Gerätehandbuch.
2. Starten Sie den PC und öffnen Sie die BIOS-Einstellungen.

   Bei Phoenix BIOS: Advanced\Plug & Play O/S auf [NO] einstellen.

   Bei anderen BIOS entsprechenden Eintrag prüfen z. B. PNP OS auf [NO].

   Nach dem Start des Betriebssystems erkennt der Hardware-Assistent die neue Hardware und fragt den Treiber ab.
3. Installieren Sie den ETCPx-Treiber von der ETC-MMI-CD (...\wdm2000\ETCPC.inf)
4. Starten Sie den PC neu.

2.7 ETC-MMI starten

1. Starten Sie das ETC-MMI über **<Start> → Programme → Lenze → ETC**

ETCN011

Das ETC-MMI-Gateway wird automatisch mitgestartet. Die Applikation ist auf der Taskleiste zu sehen:

ETCN001

Hinweis!

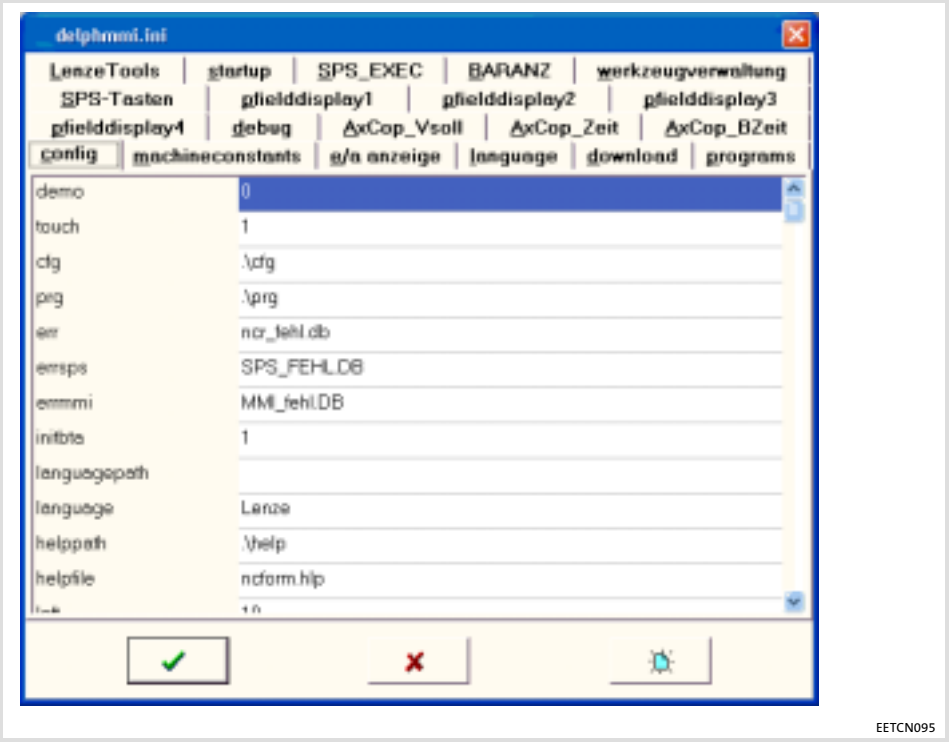
Welche Betriebsart nach dem Starten der Bedienoberfläche angezeigt wird, kann vorgegeben werden (📖 320).

Eine detaillierte Beschreibung des MMIs finden Sie im Kapitel ”ETC-MMI” (📖 280).

Eine detaillierte Beschreibung des MMI-Gateways finden Sie im Kapitel ”ETC-MMI-Gateway” (📖 270).

2.7.1 Sprache im ETC-MMI umschalten

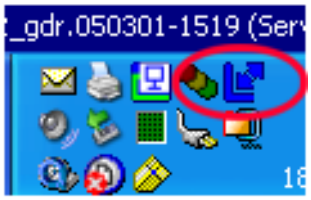
1. Drücken Sie im ETC-MMI-Fenster **<F12>** (Diagnose).
   Das MMI-Fenster der Betriebsart ”Diagnose” wird geöffnet.
2. Drücken Sie im ETC-MMI-Diagnose-Fenster **<F8>** (MMI-Konfig.).
   Das Fenster ”delphmmi.ini” wird geöffnet (📖 277).



3. Geben Sie in der Zeile *Language* die Sprache vor:
   Deutsch: ”Lenze”
   Englisch: ”Lenze_gb”
4. Schließen Sie das Fenster mit **<Enter>**.
5. Beenden Sie das ETC-MMI und starten Sie es neu.

2.7.2 Verbindung zwischen ETC-MMI und ETC herstellen

1. Klicken Sie auf der Taskleiste mit der linken Maustaste auf das ETC-MMI-Gateway-Symbol.



ETCN001

Ein Menü wird geöffnet.

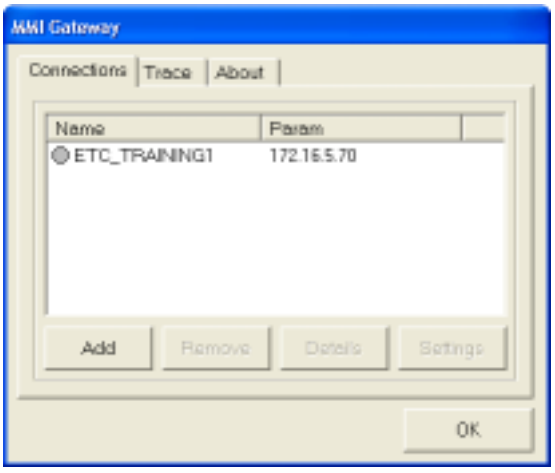
ETCN002

Settings: Konfigurationsoberfläche starten.

About: Versions- und Herstellerinformationen anzeigen.

Exit: Gateway schließen (wenn aktive Verbindungen zu einer Applikation bestehen, wird eine Warnung angezeigt).

2. Klicken Sie auf **Settings**.



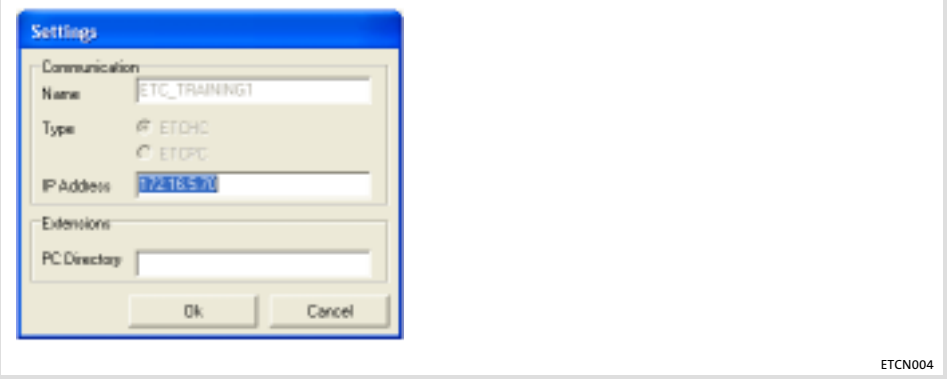
ETCN003

Connections: Liste der bereits konfigurierten Verbindungen. Ein grünes LED-Zeichen neben einem Verbindungsnamen kennzeichnet eine aktive Verbindung, über die Nachrichten und/oder zyklische Daten gesendet werden. Über die Schaltfläche **Settings** können Sie die Kommunikationsparameter der markierten Verbindung ändern (📖 272).

Trace: Bei Störungen in der Kommunikation können auf diesem Register Trace-Aufnahmen aktiviert werden (📖 275).

About: Versions- und Herstellerinformationen anzeigen.

3. Um eine neue Verbindung zu erstellen, klicken Sie auf dem Register ”Connection” auf **Add**.



4. Geben Sie einen Namen für die Verbindung ein:

   Damit eine Applikation über das ETC-MMI-Gateway mit einer Steuerung kommunizieren kann, muss jeder Verbindung ein eindeutiger Name zugewiesen werden. Beliebige Namen sind erlaubt, vergeben Sie z. B. einheitliche Namen ”ETC0”, ”ETC1”, ... oder applikationsspezifische Namen ”Rampe”, ”Lasersteuerung” usw. Ein Name kann bis 31 ASCII-Zeichen enthalten (”A ... Z”, ”a ... z” , ”0 ... 9”).

5. Wählen Sie einen Verbindungstyp:

► UDP-Verbindung für ETCHx-Steuerungen

  Ethernet-Kommunikation über DPR/UDP/IP-Protokoll.

► PCI-Verbindung für ETCPx-Steuerungen

  DPR-Kommunikation über einen WDM-Treiber (ETCPC.sys, im Lieferumfang der ETCPC).

6. Geben Sie einen Kommunikationsparameter im Feld ”IP-Adresse” ein:

► ETCHC = IP-Adresse (📖 28)

► ETCPC = Index der PCI-Karte (Indizes der PCI-Karten liegen im Bereich 0 ... 9 und werden vom Treiber automatisch zugewiesen. Die erste ETCPx hat den Index 0).

7. Schließen Sie den Dialog über **OK**.

   Ihre Eingaben werden in die Datei ”mmigtway.ini” geschrieben (📖 277). Wenn bereits eine Verbindung mit den gleichen Parametern existiert, wird der Fehler ”Gateway cannot change the parameter” gemeldet.

2.8 Antriebe über Maschinenkonstanten parametrieren

Die Eigenschaften der Antriebe müssen sowohl im Antrieb selbst als auch in der Steuerung parametriert werden.

In der Steuerung werden die Eigenschaften über Maschinenkonstanten (MKs) zugewiesen. Eine Maschinenkonstante besteht aus einem Schlüsselwort und den zugehörigen Werten; zum Beispiel "MK_VMAX 20".

Die Maschinenkonstanten sind auf dem PC in einer Textdatei mit der Endung .mk (z. B. ETCHC.mk für die ETCHC) gespeichert und werden über das ETC-MMI in die Steuerung geladen. Über einen Dialog im ETC-MMI können die Maschinenkonstanten nach dem Laden geändert werden.

2.8.1 Übersicht der wichtigsten Maschinenkonstanten

Die im Folgenden aufgeführten Maschinenkonstanten sind zwingend notwendig zum Betrieb eines Antriebs. Sie stellen bezogen auf die Antriebe nur eine Untermenge der notwendigen Parametrierung dar. Eine detailierte Beschreibung aller Maschinenkonstanten entnehmen Sie dem Kapitel "Maschinenkonstanten" (📖 188).

Stop!

Die Steuerung initialisiert über ihre Maschinenkonstanten nur die Antriebe und gibt bahnbezogene Parameter vor. Die regelungstechnischen Parameter, die Sicherheitsfunktionen der Antriebe und die Parameter der Referenzfahrt müssen mit Hilfe des GDC (Global Drive Control) im Antrieb selbst parametriert werden.

Testeinstellung

MK-Schlüsselwort	Anzahl Werte	Werte	Bedeutung
MK_TEST_OHNEMECHANIK	1	0, 1	**0:** Funktion "Achsrechner" der NC-Firmware arbeitet und berücksichtigt den Istpositionszähler des Antriebs. **1:** Funktion "Achsrechner" der NC-Firmware arbeitet, simuliert aber den Istpositionszähler des Antriebs (Istposition = Sollposition für Test der NC-Funktion ohne Mechanik).
MK_SPS_DUMMY	1	0, 1, -1	**0:** mit integrierter SPS-Funktion **1:** ohne integrierte SPS-Funktion **-1:** ohne integrierte SPS-Funktion, Freigaben werden zurückgesetzt

Hardware-Konfiguration

MK-Schlüsselwort	Anzahl Werte	Werte	Bedeutung
MK_CANDRIVES	12	-1, 0 ... 11	Zuordnung der **Achsnummer** 0 ... 11 zu der **CAN-Knotenadresse** in der Reihenfolge der CAN-Knotenadresse 1 ... 12 **-1:** Der Knotenadresse ist keine Achsnummer zugeordnet **0 ... 11:** Der Knotenadresse ist eine Achsnummer zugeordnet
MK_APPLACHSIDX	18	-1, 0 ... 11	Zuordnung der **Achsnummer** 0 ... 11 zu der **Achsbezeichnung** in der Reihenfolge der Achsbezeichnung X ,Y, Z, C, U, V, W, A, B, u, v, w, x, y, z, a, b, c **-1:** Der Achsbezeichnung ist keine Achsnummer zugeordnet **0 ... 11:** Der Achsbezeichnung ist eine Achsnummer zugeordnet
MK_ACHSENART	12		Zuordnung der **Achsenart** zu der **Achsnummer** in der Reihenfolge der Achsnummer 0 ... 11
			Bit 0 Achsenart
		xxxxxxx0	Linearachse
		xxxxxxx1	Rotationsachse
			Bit 1 Endschalter
		xxxxxx0x	HW-Endschalter berücksichtigen
		xxxxxx1x	HW-Endschalter ignorieren
			Bit 3/2 Achsenart
		xxxx00xx	Normalachse
		xxxx01xx	Spindel
		xxxx10xx	Messachse
		xxxx11xx	Spindel und Messachse
			Bit 5/4 Nur bei xxxxxxx1, Rotationsachse
		xx00xxxx	Rotationsachse mit absoluter Positionierung
		xx01xxxx	Modulo 360°-Achse - Vorzeichen gibt Richtung an
		xx11xxxx	Modulo 360°-Achse - kürzester Weg wird gefahren (nur ETCHC)
			Bit 6 Gantry-Achse
		x0xxxxxx	Normale Achse
		x1xxxxxx	Gantry-Achse (Synchronachse mit mechanischer Kopplung)
			Bit 7 Handrad
		0xxxxxxx	Normale Achse
		1xxxxxxx	Handrad

Software-Konfiguration

MK-Schlüsselwort	Anzahl Werte	Werte	Bedeutung
MK_CANOPEN_BAUDRATE	2	0, 10... 1000	1. Parameter für CAN1 2. Parameter für CAN2 **0:** kein Gerät angeschlossen **>0:** Baudrate für CAN-Open in kB
MK_DELTAT	1		Grobinterpolationstakt in [ms]

Einstellung der Achsen

MK-Schlüsselwort	Anzahl Werte	Werte	Bedeutung
MK_IMPULSE	12		Anzahl Impulse pro [MK_WEG] (nach der Vervierfachung!) in der Reihenfolge der Achsnummer 0 ... 11
MK_WEG	12		Weg in [mm] bzw. [Grad] der im Achsrechner (!) dem Wert von [MK_IMPULSE] entspricht in der Reihenfolge der Achsnummer 0 ... 11

Achsbezogene Grenzwerte

MK-Schlüsselwort	Anzahl Werte	Werte	Bedeutung
MK_MODVMAX	12		Modale Achsgeschwindigkeit (Handverfahren) [m/min] bzw. [U/min] in der Reihenfolge der Achsnummer 0 ... 11
MK_VMAX	12		Max. Achsgeschwindigkeit [m/min] bzw. [U/min] in der Reihenfolge der Achsnummer 0 ... 11
MK_BESCHL	12		Beschleunigungsrampe [m/s^2] bzw. [U/s^2] in der Reihenfolge der Achsnummer 0 ... 11
MK_BREMS	12		Bremsrampe [m/sec^2] bzw. [U/s^2] in der Reihenfolge der Achsnummer 0 ... 11
MK_T_BESCHL	12		Dämpfungszeitkonstante für Beschleunigungs- und Bremsrampen [ms] in der Reihenfolge der Achsnummer 0 ... 11

Bahnbezogene Grenzwerte

MK-Schlüsselwort	Anzahl Werte	Werte	Bedeutung
MK_VBAHNMAX	1		Max. Bahngeschwindigkeit [m/min]
MK_BAHNBESCHL	1		Beschleunigungsrampe [m/s^2]
MK_BAHNBREMS	1		Bremsrampe [m/s^2]
MK_T_BAHNBESCHL	1		Dämpfungszeitkonstante für Brems- und Beschleunigungsrampen [ms]

2.8.2 Maschinenkonstanten-Datei ETCxC.mk

In der Steuerungsvariante ”mit MMI” wird die Datei ETCxC.mk beim Start des ETC-MMI in die Steuerung ETCxC geladen. In der Steuerungsvariante ”Stand-Alone”-Variante wird erkannt, dass die Maschinenkonstanten schon geladen sind.

Das folgende Beispiel von Maschinenkonstanten ist ein Ausschnitt aus der Datei ETCxC.mk mit voreingestellten Maschinenkonstanten, die für den konkreten Anwendungsfall angepasst werden müssen.

Testeinstellung

MK-Schlüsselwort	Anzahl Werte	Werte
MK_TEST_OHNEMECHANIK	1	0
MK_SPS_DUMMY	1	0

Hardware-Konfiguration

MK-Schlüsselwort	Anzahl Werte	Werte
MK_CANDRIVES	12	0, 1, 2, 3, -1, -1, -1, -1, -1, -1, -1, -1
MK_APPLACHSIDX	18	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17
MK_ACHSENART	12	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

Software-Konfiguration

MK-Schlüsselwort	Anzahl Werte	Werte
MK_CANOPEN_BAUDRATE	2	500, 1000
MK_DELTAT	1	2

Einstellung der Achsen

MK-Schlüsselwort	Anzahl Werte	Werte
MK_IMPULSE	12	65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536
MK_WEG	12	10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10

Achsbezogene Grenzwerte

MK-Schlüsselwort	Anzahl Werte	Werte
MK_MODVMAX	12	10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10
MK_VMAX	12	20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20
MK_BESCHL	12	2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2
MK_BREMS	12	2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2
MK_T_BESCHL	12	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

Bahnbezogene Grenzwerte

MK-Schlüsselwort	Anzahl Werte	Werte
MK_VBAHNMAX	1	20
MK_BAHNBESCHL	1	2
MK_BAHNBREMS	1	2
MK_T_BAHNBESCHL	1	0

2.8.3 Hinweise zum Laden der MK-Datei in die Steuerung

Achten Sie darauf, dass die Anzahl der Parameter in der Datei der Maschinenkonstanten mit der Anzahl der maximalen Achsen (12) im Betriebssystem der Steuerung übereinstimmt.

Beim Laden der Datei der Maschinenkonstanten in die Steuerung sind 3 Fälle möglich:

- Anzahl der Parameter in der Datei **ist gleich** der maximalen Anzahl im Betriebssystem.

  Die Parameter überschreiben die gespeicherten Maschinenkonstanten in der Steuerung.

- Anzahl der Parameter in der Datei **ist größer** als maximale Anzahl im Betriebssystem.

  Die Parameter überschreiben die gespeicherten Maschinenkonstanten in der Steuerung bis zu der maximalen Anzahl. Für jeden nicht übernommenen Parameter der Datei wird eine Fehlermeldung erzeugt.

- Anzahl der Parameter in der Datei **ist kleiner** als maximale Anzahl im Betriebssystem.

  Die Parameter überschreiben die gespeicherten Maschinenkonstanten in der Steuerung. Die fehlenden Maschinenkonstanten bis zu der maximalen Anzahl werden durch die in der Steuerung bereits vorhandenen Einträge ergänzt.

Hinweis!

Wenn sich die Steuerung in einem Fehlerzustand befindet (Anzeige im ETC-MMI in der Betriebsart EINRICHTEN: ”NCR: FEHLER”), werden die geladenen Maschinenkonstanten nicht in die Steuerung übernommen.

Erst im Steuerungszustand ”Idle” (Anzeige im ETC-MMI in der Betriebsart EINRICHTEN: ”NCR: STEHT”) sind die Maschinenkonstanten korrekt übernommen.

2.8.4 Beispiel für das Anpassen einer Maschinenkonstanten-Datei

Die Antriebskonfiguration umfasst 3 Antriebe mit folgenden Eigenschaften:

	Antrieb 1	Antrieb 2	Antrieb 3
Achsennummer	0	1	1
CAN-Knotenadresse	8	7	3
CAN-Baudrate	500 kB		
Achsbezeichnung	C	X	(X')
Achsenart	Rotationsachse	Linearachse mit Handrad	Gantryachse zu Antrieb 2 mit gleichen Eigenschaften
Auflösung Impule	65536	65536	wie Antrieb 2

Die Datei der Maschinenkonstanten muss dafür wie folgt aussehen:

Testeinstellung

MK-Schlüsselwort	Anzahl Werte	Werte
MK_TEST_OHNEMECHANIK	1	0
MK_SPS_DUMMY	1	0

Hardware-Konfiguration

MK-Schlüsselwort	Anzahl Werte	Werte
MK_CANDRIVES	12	-1, -1, **1**, -1, -1, -1, **1**, **0**, -1, -1, -1, -1
MK_APPLACHSIDX	18	**1**, -1, -1, **0**, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1
MK_ACHSENART	12	**1**, **192**, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

Software-Konfiguration

MK-Schlüsselwort	Anzahl Werte	Werte
MK_CANOPEN_BAUDRATE	2	500, 1000

Einstellung der Achsen

MK-Schlüsselwort	Anzahl Werte	Werte
MK_IMPULSE	12	65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536, 65536

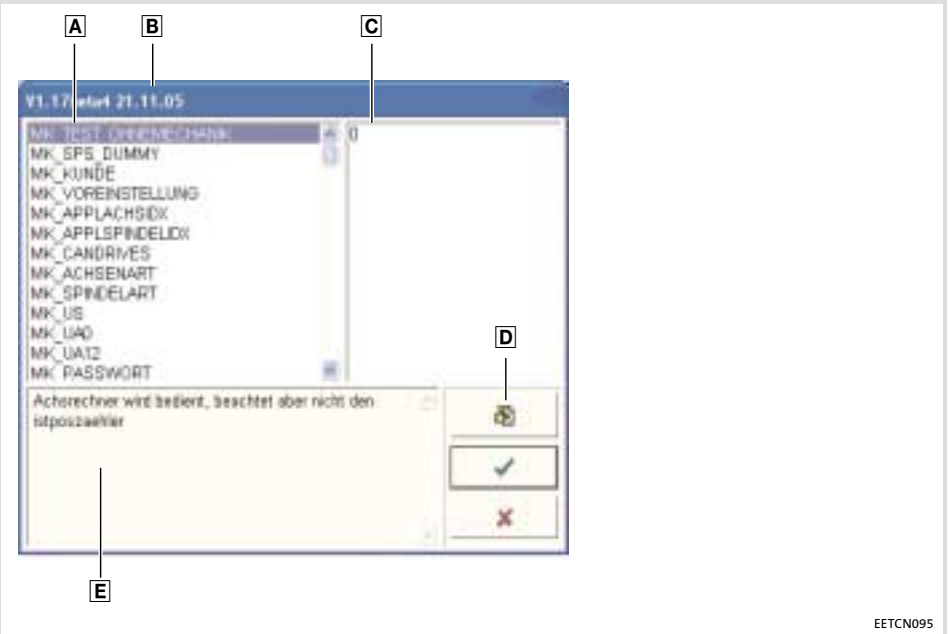
Erläuterung

Um eine Synchronachse zu definieren wird in MK_CANDRIVES die gleiche Achsnummer an 2 CAN-Knotenadressen eingetragen. Damit wird eine Zwangskopplung der Achsen der beiden CAN-Knotenadressen erzeugt. Die CAN-Achse mit der kleineren Knotenadresse ist automatisch die Führungsachse, die Achse mit der größeren Knotenadresse und der gleichen Achsnummer ist die Folgeachse (Synchronachse).

Die Gantry-Achse (X', in MK_ACHSENART) ist ein Sonderfall der Synchronachse. Hier handelt es sich um eine mechanische Zwangskopplung von 2 Achsen. Die Gantry-Achse wird daher in MK_CANDRIVES nicht als eigene Achse eingetragen, sondern nur als Bezug auf die Achse X.

2.8.5 Maschinenkonstanten im ETC-MMI anpassen

1. Drücken Sie im ETC-MMI-Fenster **<F12>** (Diagnose).
   Das MMI-Fenster der Betriebsart "Diagnose" wird geöffnet.
2. Drücken Sie dort **<F6>** (Maschinenkonst.).
3. Drücken Sie erneut **F6** (Akt. MKs ändern).
   Die aktuellen Maschinenkonstanten werden geladen.



A: Verfügbare Maschinenkonstanten

B: Datum und Version der NC-Firmware

C: Wert der markierten Maschinenkonstante

D: Geänderten Wert übernehmen

E: Kurzbeschreibung der markierten Maschinenkonstante

4. Markieren sie im Feld der verfügbaren Maschinenkonstanten A diejenige, die Sie anpassen möchten.
5. Um den Wert der markierten Maschinenkonstante zu ändern, drücken Sie die **<Leertaste>** und geben den neuen Wert im Feld C ein.
6. Zum Übernehmen des Wertes, drücken Sie die Taste D.
7. Wenn Sie alle Anpassungen durchgeführt haben, drücken Sie **<Enter>**.
   Die Änderungen werden zur ETC gesendet und in der MK-Datei aktualisiert.

Testeinstellung

Hinweis!

Der Betrieb des CNC-Programms ist grundsätzlich auch möglich:

- ohne angeschlossene Mechanik und Antriebe.
  Dies wird erreicht durch Setzen der Maschinenkonstanten MK_TEST_OHNEMECHANIK=1.
- ohne ein SPS-Programm.
  Dies wird erreicht durch Setzen der Maschinenkonstanten MK_SPS_DUMMY=1.

Für die folgenden Tests mit Mechanik müssen die oben genannten Maschinenkonstanten wie folgt eingestellt werden:

MK-Schlüsselwort	Anzahl Werte	Werte
MK_TEST_OHNEMECHANIK	1	0
MK_SPS_DUMMY	1	1

2.8.6 Parameter der Antriebe überprüfen

Nach Anpassen der Maschinenkonstanten können die angeschlossenen Antriebe in der Diagnose über ihren Schleppabstand überprüft werden.

1. Drücken Sie im ETC-MMI-Fenster **<F12>** (Diagnose).
   Das MMI-Fenster der Betriebsart "Diagnose" wird geöffnet.

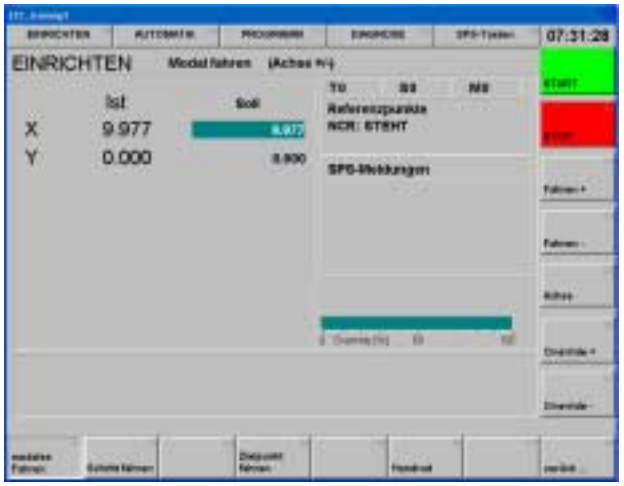
ETCN012

Richtig parametrierte Antriebe zeigen einen wechselnden Schleppabstand um den Nullpunkt.

2.8.7 Antriebe im Tippbetrieb testen

Nach der Anpassung der Maschinenkonstanten müssen die Antriebe im Tippbetrieb getestet werden. Kontrollieren Sie dabei, dass sich die konfigurierten Antriebe gemäß den Vorgaben verhalten.

1. Drücken Sie im ETC-MMI-Fenster **<F9>** (Einrichten).
   Das MMI-Fenster der Betriebsart ”Einrichten” wird geöffnet.
2. Drücken Sie dort **<F2>** (Handfahren).
3. Drücken Sie im Untermenü auf **<F1>** (Modales Fahren).
   Ein einzelner Antrieb ist ausgewählt. Der ausgewählte Antrieb ist am grün unterlegten Sollwert zu erkennen.



ETCN011

4. Drücken Sie im Untermenü abwechselnd auf **<S3>** (Fahren +) und **<S4>** (Fahren -).
   Der ausgewählte Antrieb dreht sich im Tippbetrieb in positive bzw. negative Richtung.
5. Um die Geschwindigkeit des Antriebes zu ändern, drücken Sie **<S6>** (Override +) bzw. **<S7>** (Override -).
6. Um die nächste Achse zum Verfahren auszuwählen, drücken Sie **<S5>** (Achse).
   Die grüne Sollwertanzeige schaltet um.

2.9 CNC-Programmierung nach DIN 66025

Die folgende Beschreibung von Funktionen nach DIN 66025 ist ein Auszug aus dem Kapitel ”CNC-Programmierung” (📖 85).

2.9.1 G-Funktionen

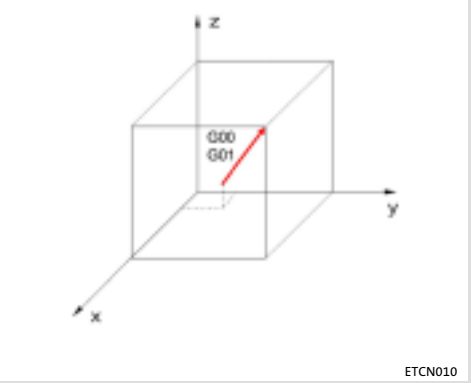
G-Funktionen definieren geometrische Wegbedingungen für den Betrieb der Achsen. Ein DIN-Satz mit einer G-Funktion hat grundsätzlich folgenden Aufbau:

Dem Buchstaben ”G” folgt die Nummer der G-Funktion. Daran anschließend folgen die Parameter, die jeweils durch einen Buchstaben und dem dazugehörigen Wert gebildet werden.

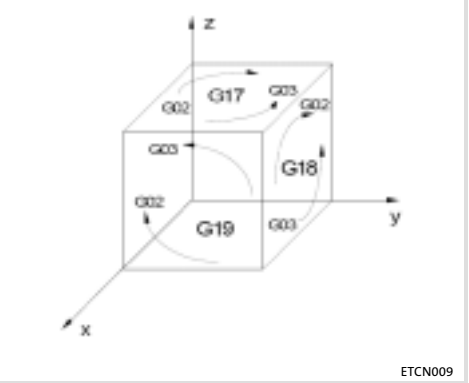
G (Nummer) [Parameter (Wert)] [Parameter (Wert)] ...

Im Folgenden einige Beispiele von G-Funktionen:

G	Bedeutung	Parameter	Beispiel mit Parameter und Werten
00	Geradeninterpolation im Eilgang	AXES R D F E L	G1 X0 Z10 C0 F2
01	Geradeninterpolation		
02	Kreisinterpolation im Uhrzeigersinn	AXES D I J K R F E L	G3 X40 Y40 R100 D-60 K2
03	Kreisinterpolation im Gegenuhrzeigersinn		
17	Ebenenauswahl XY zur Kreisinterpolation	--	G17
18	Ebenenauswahl XZ zur Kreisinterpolation	--	G18
19	Ebenenauswahl YZ zur Kreisinterpolation	--	G19



ETCN010



ETCN009

Parameter für G00, G01

Parameter	Bedeutung
AXES	Zielpunktkoordinaten der linearen Achsen X, Y, Z, A, B, C, U, V, W, x, y, z, a, b, c, u, v, w Anmerkung: In einer G-Funktion dürfen nur die Achsen X, Y, Z, A, B, C, U, V, W oder die Achsen x, y, z, a, b, c, u, v, w benutzt werden.
D	Max. Bahnabweichung im Zielpunkt zum Eckenverschleifen mit der nachfolgenden Linearinterpolation.
R	Radius, mit dem die nachfolgende Linearinterpolation (G0, G1) verbunden werden soll.
F	Eilganggeschwindigkeit (G00) bzw. Vorschubgeschwindigkeit (G01) auf der Bahn (G31) bzw. jeder Achse (G30), auch für alle nachfolgenden Linearinterpolationen.
E, L	Auswahl der Vorschubgeschwindigkeit über Drehzahl (E) und Schrittweite (L) F=E*L.

Parameter für G02, G03

Parameter	Bedeutung
AXES	Zielpunktkoordinaten der drei linearen Hauptachsen (Default X,Y,Z), sowie Start- bzw. Zielpunktkoordinaten der drei rotativen Nebenachsen (Default A,B,C).
D	Radiusänderung bei Erreichen des Zielpunktes.
I	Mittelpunktkoordinate der ersten Hauptachse (X) oder Anzahl zusätzlicher Vollkreise.
J	Mittelpunktkoordinate der zweiten Hauptachse (Y) oder Anzahl zusätzlicher Vollkreise.
K	Mittelpunktkoordinate der dritten Hauptachse (Z) oder Anzahl zusätzlicher Vollkreise
R	Interpolationsradius
F	Bahngeschwindigkeit
E, L	Auswahl der Vorschubgeschwindigkeit über Drehzahl (E) und Schrittweite (L) F=E*L.

2.9.2 M-Funktionen

M-Funktionen definieren Maschinenanweisungen. Sie werden in der CNC bearbeitet und außerdem an die SPS gesendet. In der SPS werden sie ausgewertet und lösen dort die programmierte Funktion im Zusammenspiel mit der CNC aus.

M-Funktionen haben keine Parameter. Die Koordination zwischen der SPS und der CNC erfolgt über die Schnittstellensignale SPS-NC (📖 237).

Im Folgenden einige Beispiele von festen M-Funktionen. Alle weiteren M-Funktionen stehen dem Programmierer zur freien Verfügung.

Parameter	Bedeutung
14	Konturanfang, schaltet die Aufsummierung der Bahnlänge im Parameterfeld P551 ein. Bei Verwendung von M14 in einem Online-Programm (Programm mit Nachladen von Programmteilen) wird die Ausführung des Programms solange blokkiert, bis sich das Konturende (M15/M16/M21) im Speicher befindet.
15	Konturende, schaltet die Aufsummierung der Bahnlänge in P551 aus (Beginn der Leerfahrt).
30	Programmende

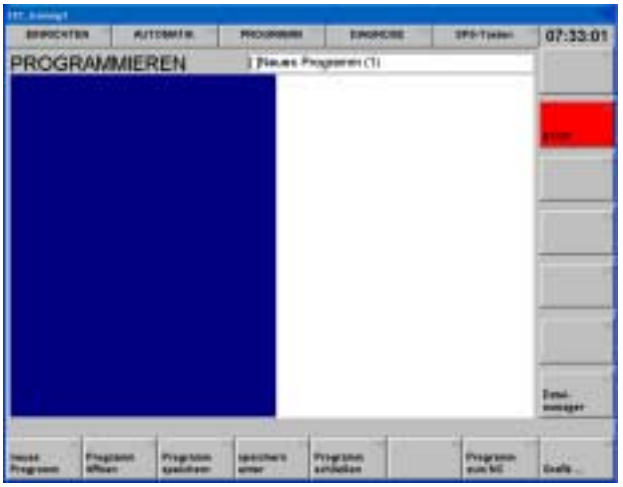
2.10 CNC-Beispielprogramm erstellen

2.10.1 Texteditor im ETC-MMI aufrufen

1. Starten Sie ggf. das ETC-MMI über **<Start> → Programme → Lenze → ETC**
   Das ETC-MMI-Fenster wird geöffnet.

ETCN011

2. Drücken Sie im ETC-MMI-Fenster **<F11>** (Programm).
   Das MMI-Fenster in der Betriebsart ”Programmierung” wird geöffnet. Es zeigt den Texteditor für die Eingabe des CNC-Programms.
3. Drücken Sie **<F8>** (Grafik) und anschließend **<F7>** (Grafik ein/aus).
   Das Fenster wird geteilt. Es zeigt links den Texteditor und rechts ein Anzeigefeld für die grafische Darstellung (Grafikbereich). Im Texteditor blinkt links oben der Cursor.



ETCN014

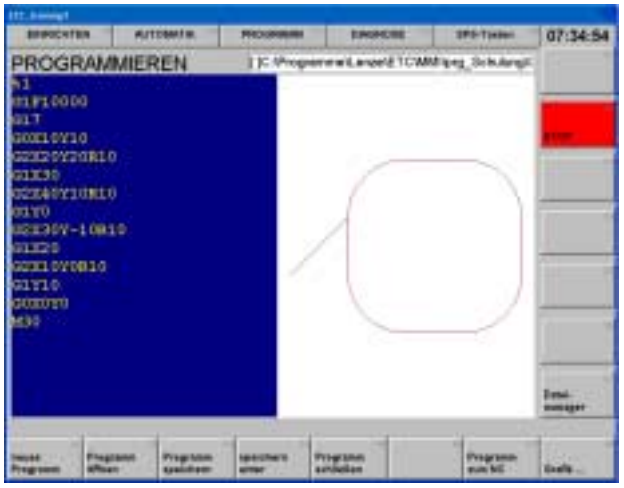
2.10.2 CNC-Programm eingeben und speichern

Wir erstellen ein Programm für eine Kontur mit abgerundeten Ecken.

Ausgangspunkt: Der Cursor blinkt oben rechts im Texteditor und der Grafikbereich wird angezeigt.

1. Geben Sie das CNC-Programm gemäß folgender Tabelle in den Texteditor ein:

Zeile	Eingabe im Texteditor	Bedeutung	Anzeige im Grafikbereich
1	%1	Programmnummer festlegen	
2	G1 F10000	Fahrgeschwindigkeit festlegen	
3	G17	Ebene festlegen	
4	G0 X10 Y10	Zum Ausgangspunkt mit Eilgang fahren	erzeugt eine Gerade vom Nullpunkt zum Ausgangspunkt
5	G2 X20 Y20 R10	Ein Teilstück der quadratischen Kontur abfahren	erzeugt einen Viertelkreis
6	G1 X30		ergänzt eine Gerade
7	G2 X40 Y10 R10		ergänzt einen Viertelkreis
8	G1 Y0		ergänzt eine Gerade
9	G2 X30 Y-10 R10		ergänzt einen Viertelkreis
10	G1 X20		ergänzt eine Gerade
11	G2 X10 Y0 R10		ergänzt einen Viertelkreis
12	G1 Y10		ergänzt eine Gerade zu einer geschlossenen quadratischen Kontur
13	G0 X0 Y0	Zum Nullpunkt mit Eilgang fahren	
14	M30	Ende des Programms	



ETCN015

2. Drücken Sie **<F4>** (Speichern unter).
3. Geben Sie als Programmname z. B. **test.din** ein.
4. Drücken Sie **<Enter>**.

2.10.3 CNC-Programm in die Steuerung laden und starten

Programm in die Steuerung laden

Ausgangspunkt: SPS-Programm wird im Editor angezeigt.

1. Drücken Sie **<F7>** (Programm zum NC).
2. Wählen Sie das Programm; z. B. **test.din**.
3. Drücken Sie **<Enter>**.
   Das Programm wird auf die Steuerung ETCxC übertragen.

Programm in der Steuerung starten

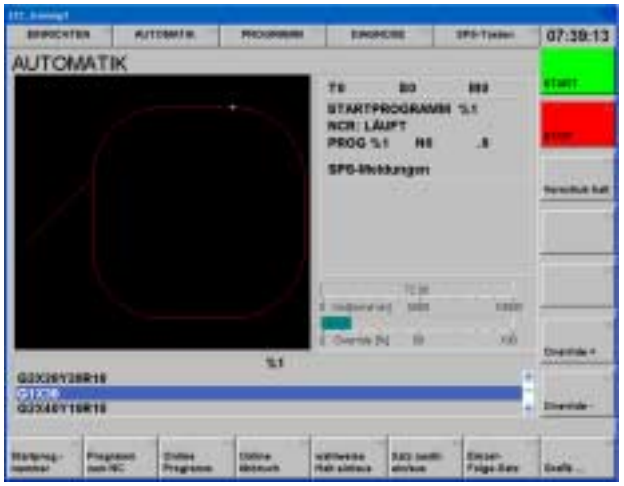
Ausgangspunkt: Der Cursor blinkt oben rechts im Texteditor und der Grafikbereich wird angezeigt. In der Steuerung ist das Programm (z. B. **test.din**.) geladen.

Hinweis!

Der Betrieb des CNC-Programms ist grundsätzlich auch ohne angeschlossene Mechanik und Antriebe möglich und auch ohne ein SPS-Programm. Dazu sind folgende Maschinenkonstanten zu setzen:

- ► MK_TEST_OHNEMECHANIK=1
- ► MK_SPS_DUMMY=1

1. Drücken Sie **<S1>** (Start).
   Die Antriebe starten und die Kontur des Rechtecks wird abgefahren (ohne eingeschaltete Grafik ändern sich die Zahlen der Soll- und Istpositionen). Die einzelnen Befehle werden im unteren Teil des Fensters angezeigt.



ETCN016

   Die gezeichnete Gerade vom Nullpunkt (0/0) zum Anfangspunkt (10/10) wird nur dann abgefahren, wenn der Antrieb auch im Nullpunkt steht. Andernfalls fährt die Steuerung die Antriebe auf einer Geraden vom aktuellen Punkt zum Anfangspunkt.
2. Um die Bahngeschwindigkeit zu ändern, drücken Sie **<S6>**/**<S7>** (Override +/-).
3. Um die Antriebe zu stoppen, drücken Sie **<S2>** (Stop).

2.10.4 CNC-Programm erweitern

Das Beispiel-Programm (”test.din”) fährt eine kreisförmige Kontur ab. Das eigentliche Ziel eines CNC-Programms ist, beim Abfahren einer Kontur ein Werkzeug einzuschalten. Das Beispiel-Programm wird daher um die entsprechenden M-Funktionen M14 und M15 erweitert, die beim Abfahren der Kontur das Werkzeug absenken bzw. anheben. Die M-Funktionen werden in einem getrennten SPS-Programm programmiert.

M-Funktion	Soll-Funktion	Programmierung in einem getrennten SPS-Programm
M14	Werkzeug absenken	Ein Ausgang zum Absenken des Werkzeugs soll gesetzt werden. Weiterschalten auf den nächsten Befehl erst nach Ansprechen eines unteren Endschalters.
M15	Werkzeug anheben	Ein Ausgang zum Anheben des Werkzeugs soll gesetzt werden. Weiterschalten auf den nächsten Befehl erst nach Ansprechen eines oberen Endschalters.

Ausgangspunkt: Der Cursor blinkt oben rechts im Texteditor und der Grafikbereich wird angezeigt. In der Steuerung ist das Programm (z. B. **test.din**.) geladen.

1. Drücken Sie **<F2>** (Programm öffnen).
2. Wählen Sie das Programm (z. B. **test.din**) und drücken Sie **<Enter>**
3. Drücken Sie **<F7>** (Übernahme).
4. Fügen Sie die M-Funktionen gemäß nachstehender Tabelle ein.

Eingabe im Texteditor	Bedeutung
%1	
G1 F10000	
G17	
M15	Werkzeug anheben
G0 X10 Y10	
M14	Werkzeug absenken
G2 X20 Y20 R10	
G1 X30	
G2 X40 Y10 R10	
G1 Y0	
G2 X30 Y-10 R10	
G1 X20	
G2 X10 Y0 R10	
G1 Y10	
M15	Werkzeug anheben
G0 X0 Y0	
M30	

5. Drücken Sie **<F3>** (Programm speichern).

ETC-MMI beenden

1. Drücken Sie **<F9>** (Einrichten).
2. Drücken Sie **<F8>** (Programm beenden).

2.11 ETC-SPS-Programmierung mit CoDeSys

Das Werkzeug ETC-CoDeSys ist eine komplette Entwicklungsumgebung zum Erstellen und Testen von SPS-Programmen für die ETC.

Es basiert auf dem weit verbreiteten Programmpaket CoDeSys mit den spezifischen Erweiterungen für die ETC-Steuerungen.

Hinweis!

Weitere Informationen erhalten Sie im Kapitel ”SPS-Programmierung” (📖 329).

2.11.1 CoDeSys installieren

1. Legen Sie die CoDeSys-Setup-CD in Ihr CD-ROM-Laufwerk.

   Wenn das Autostart-Feature des CD-ROM-Laufwerks in Windows aktiviert ist, startet das Setup automatisch.
2. Sollte das Setup nicht automatisch starten:
   – Wählen Sie im Startmenü den Menüpunkt ”Ausführen”.
   – Geben Sie in der Befehlszeile den Laufwerksbuchstaben Ihres CD-ROM-Laufwerks gefolgt von ”setup.exe” ein (z. B. ”d:\setup.exe”) und bestätigen Sie mit ”OK”.
3. Bei der Auswahl des Zielverzeichnisses generieren Sie bitte ein neues Verzeichnis ”CoDeSys” unter ”c:\programme\Lenze\ETC”.
4. Folgen Sie den Anweisungen des Setup-Programms.
5. Ändern Sie den Installationspfad in ”c:\programme\lenze\etc\CoDeSys2”.

Stop!

Die Datei ”codesys.ini” im Installationsverzeichnis kann nach der Installation von CD schreibgeschützt sein. Der Schreibschutz muss aufgehoben werden, da sonst die CoDeSys-Einstellungen nicht gespeichert werden.

2.11.2 Steuerung im ETC-CoDeSys konfigurieren

Dieses Kapitel beschreibt die generelle Vorgehensweise bei der Steuerungskonfiguration als Voraussetzung zum Erstellen eines Programms mit dem ETC-CoDeSys am Beispiel einer ETCHC.

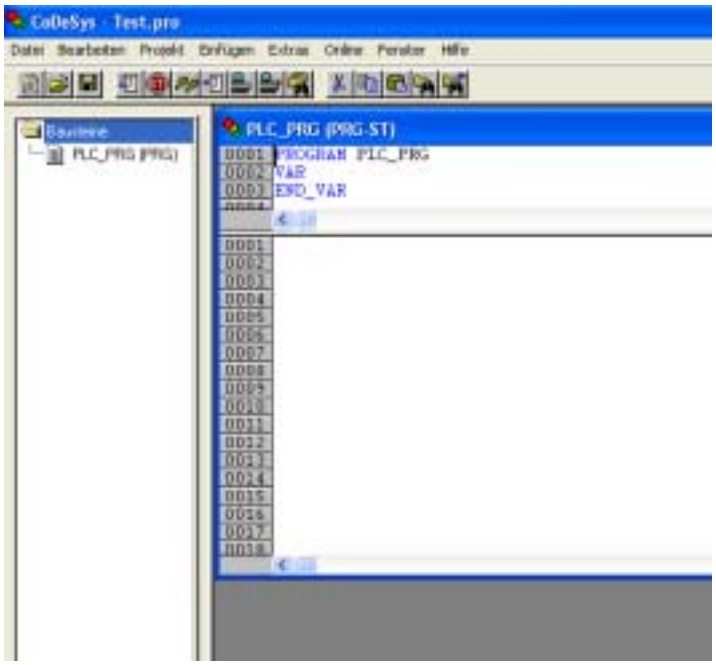
Dieses Kapitel kann übersprungen werden, wenn das Beispielprogramm (📖 58) aufgerufen wird, da die beschriebenen Einstellungen dort schon vorhanden sind.

ETC-CoDeSys starten und neues Projekt anlegen

1. Starten Sie das ETC-CoDeSys über **<Start> → Programme → Lenze → CoDeSys**.
2. Bei Bedarf können Sie die Sprache im CoDeSys umschalten:
   – **Projekt → Optionen → Arbeitsbereich → Sprache = englisch**
   – **Project → Options → Desktop → Language = german**
3. Erstellen Sie ein neues Projekt über **Datei → Neu**.
4. Wählen Sie im Dialog "Zielsystemeinstellungen" die entsprechende Steuerung.
5. Akzeptieren Sie die Voreinstellung.
   Ein Dialog für den ersten Baustein erscheint. Der Name ist frei wählbar (z. B. PLC_PRG). Als erster Baustein wird in der Voreinstellung ein Programm in der Sprache ST (strukturierter Text) erstellt.

ETCM019

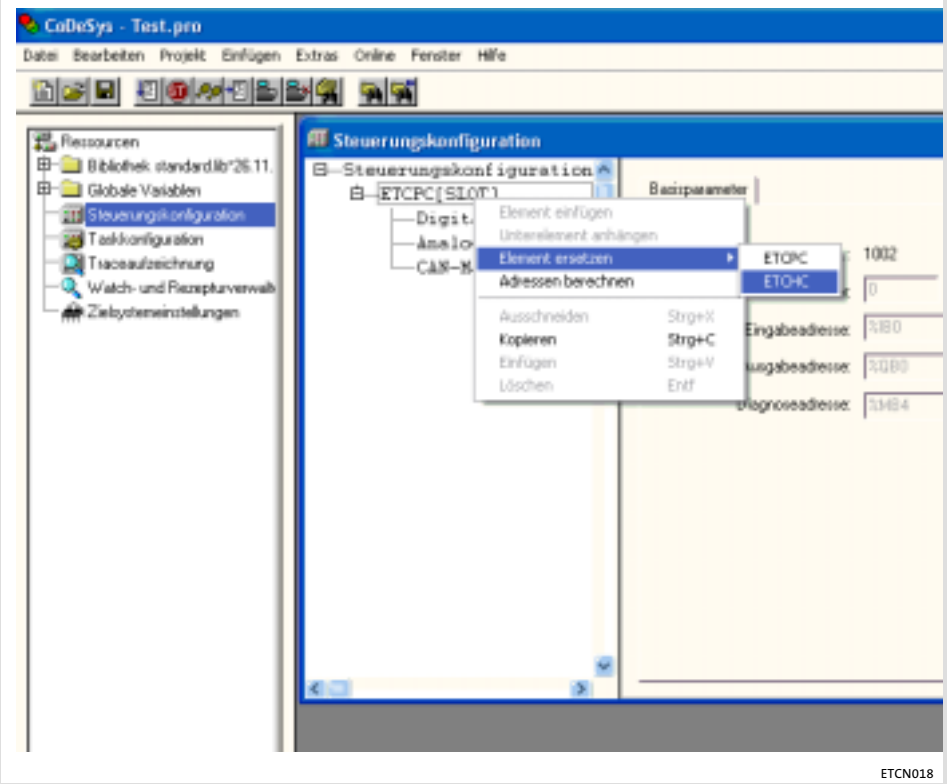
6. Drücken Sie **<OK>**
7. Speichern Sie das neue Projekt über **Datei → Speichern**.



ETCM020

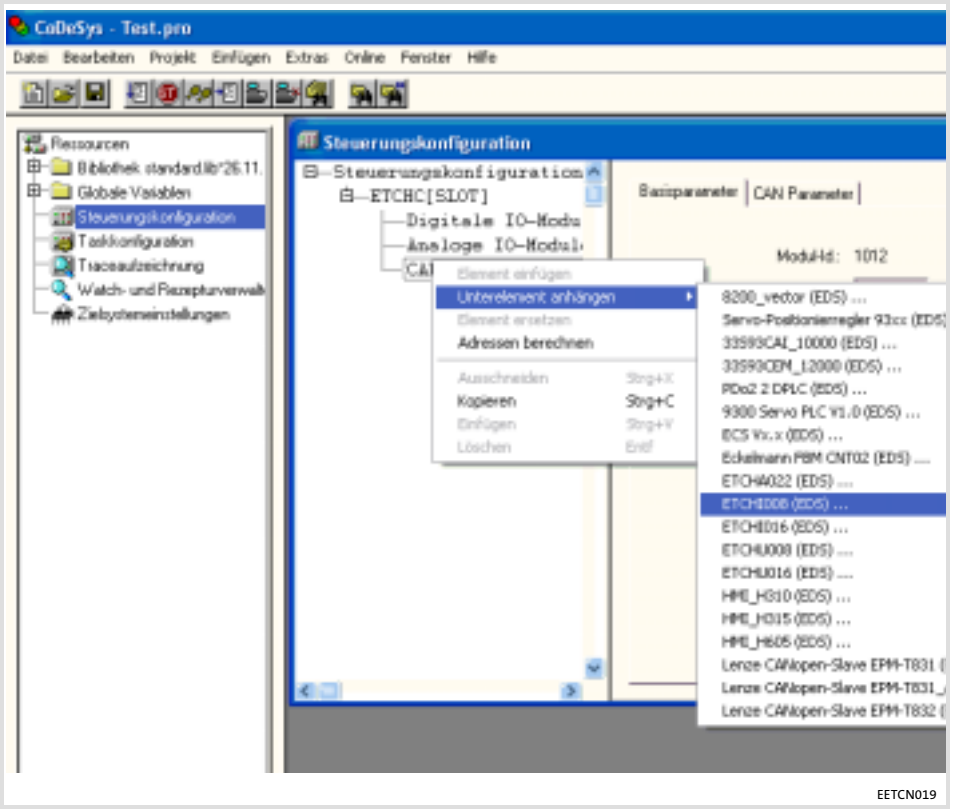
Steuerungskonfiguration im CoDeSys anwählen

1. Klicken Sie im linken, unteren Fensterbereich auf den Reiter ”Resourcen”.
2. Wählen Sie ”Steuerungskonfiguration”.
3. Öffnen Sie im rechten Fensterbereich die Steuerungskonfiguration. Der Eintrag ”ETCPC [Slot]” erscheint.
4. Klicken Sie auf ”ETCPC [Slot]” mit der rechten Maustaste.
5. Wählen Sie im Kontextmenü den Menüpunkt **Element ersetzen → ETHC**.



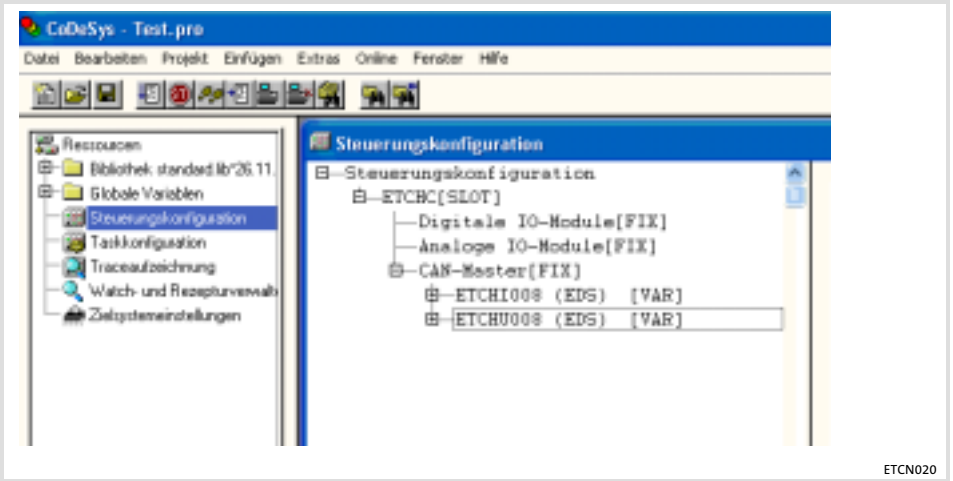
Steuerung (CAN-Master) konfigurieren

1. Öffnen Sie den Eintrag **ETCHC [Slot]** indem Sie auf ”+” klicken.
2. Klicken Sie mit der rechten Maustaste auf **CAN-Master (Fix)**.
3. Wählen Sie im Kontextmenü den Eintrag **Unterelemente anhängen** und im Untermenü das Modul, das Sie anhängen möchten.



4. Wiederholen Sie den letzten Schritt für alle Module, die Sie anhängen möchten in der Reihenfolge ihrer Position auf der Hutschiene.

   Nach der Installation der Module sieht die Konfiguration wie folgt aus:



Hinweis!

Neue Module werden an das Ende der Liste gestellt. Um ein neues Modul vor ein bestehendes einzufügen, das bestehende anwählen und über die rechte Maustaste mit **Element einfügen** einfügen.

Ein Modul kann aus der Liste über die rechte Maustaste und **Löschen** entfernt werden.

Beschreibung des Moduls in der EDS-Datei

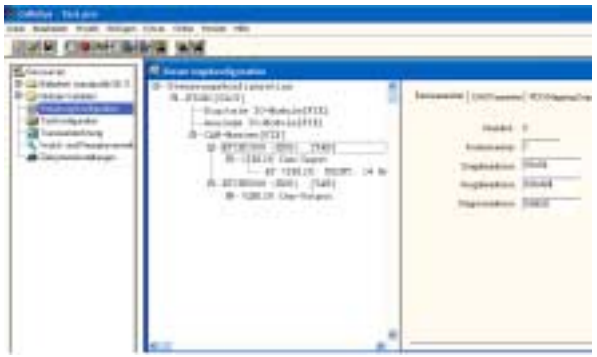
Jedes in der Steuerung zu konfigurierende Modul benötigt eine EDS-Datei. Die zum Modul gehörige EDS-Datei muss in dem jeweiligen Target-Verzeichnis der ausgewählten Steuerung (hier Lenze) liegen.

Eine EDS-Datei enthält Einträge von Schlüsselworten, die mit den entsprechenden Werten die Eigenschaft des Moduls beschreiben.

Bei Modulen von Fremdherstellern müssen die entsprechenden EDS-Dateien in das Target-Verzeichnis der Steuerung kopiert werden.

Module parametrieren

Nach Anklicken der einzelnen Module sieht das Bild wie folgt aus:



ETCN021

Dem Programm muss mitgeteilt werden, an welchen Adressen des Speicherbereichs die Eingabe- und Ausgabedaten der Module liegen. Die Voreinstellung im Programm sind Byte-Adressen (*%IB*, *%QB*), die in Wort-Adressen (*%IW*, *%QW*) geändert werden müssen, da das Prozessabbild der ETC wortweise organisiert ist. Der Anfang des Adressbereichs liegt dabei bei Wort 64. Bei der Änderung in Wort-Adressen ist wie folgt vorzugehen:

► Beginnend mit dem ersten Modul in der Liste (also hier ETCHI008) in der Reihenfolge der Module (also ETCHU008) den Typ der Variablen *%IB*, *%QB* in *%IW*, *%QW* ändern. Dabei erhält das erste Modul als Adresse Wort 64 und die folgenden Module erhalten als Wortadresse Wort 64 plus der Byte-Adresse minus 128.

Modul	angezeigte Eingabe-adresse	angezeigte Ausgabe-adresse		Eingabe Eingabeadresse	Eingabe Ausgabeadresse	Diagnose-adresse
ETCHI008	%IB128	%QB128	→	%IW64	%QW64	unverändert
ETCHU008	%IB129	%QB128	→	%IW65	%QW64	
Nächstes Modul	%IB129	%QB130	→	%IW65	%QW66	

Die Diagnoseadresse muss nicht geändert werden.

Hinweis!

Die beschriebene Methode der Adressvergabe vermeidet Adressüberschneidungen der einzelnen Module, vergibt aber die Adressen nicht platzoptimal.

Für eine platzoptimale Vorgabe der Adressen folgender Module müssen diese errechnet werden, worauf hier aus Vereinfachungsgründen verzichtet wird.

Adressen automatisch

In der Steuerungskonfiguration sollte das Häkchen ”Adressen automatisch” nicht gesetzt werden, da sonst CoDeSys die Adressen bei Änderungen der Steuerungskonfiguration neu vergibt.

ETCN022

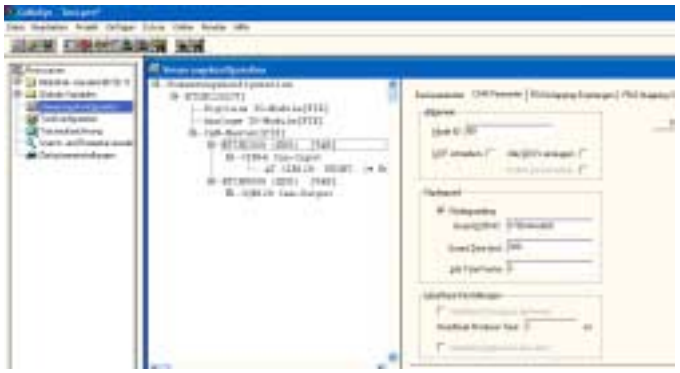
CAN-Adresse eingeben

Jedes Modul hat eine eigene CAN-Adresse (Node-ID). Eine Adresse setzt sich aus der typspezifischen Basisadresse und der individuell eingestellten Adresse am Hex-Schalter auf der Frontseite zusammen.

ETCM029

Wenn für die eingesetzten Feldbus-Module der HEX-Schalter auf Null eingestellt ist (Pfeil zeigt nach links), müssen bei den einzelnen Modulen folgende resultierende Node-ID eingetragen werden:

Modul	CAN-Basisadresse	Eingestellt am Hex-Schalter	Resultierende Node-ID
ETCHI0xx	80_{dez}	0	80_{dez}
ETCHU0xx	16_{dez}	0	16_{dez}
ETCHA022	48_{dez}	0	48_{dez}



ETCN023

Nodeguarding

Ist die Option **Nodeguarding** aktiviert, wird im Intervall, das bei **Guard Time** in Millisekunden angegeben ist, eine Nachricht an das Modul verschickt. Meldet sich das Modul daraufhin nicht mit einer Antwort und ist die Anzahl der Versuche (**Life Time Factor**) erreicht, so gilt das Modul als nicht NOK (ausgefallen oder nicht vorhanden).

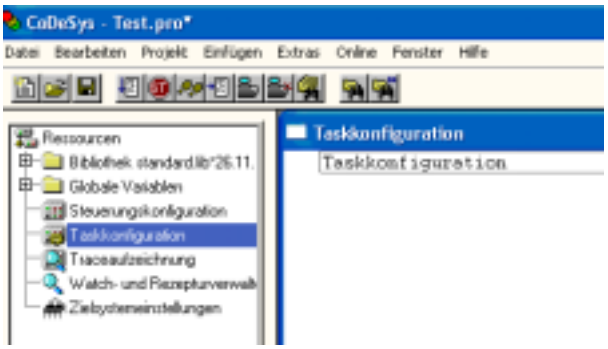
Der Status des Moduls wird in der Systemvariable der Steuerung DB1_NC2SPS_COPMODUL_VORHANDEN_AW hinterlegt (📖 237).

Die Standard-Einstellung ist bei allen Modulen **Guard Time = 500** und **Life Time Factor = 3**. Diese Einstellung bedeutet ein Erkennen eines Ausfalls nach 500 ms x 3 = 1,5 s. Wenn keine Guard Time und kein Life Time Factor angegeben ist (= 0), wird das Modul nicht überwacht.

Weitere Einstellungen sind nicht erforderlich.

Taskkonfiguration

Eine Taskkonfiguration ist nur notwendig, falls Systemvariablen von Datenbaustein_0 benutzt werden (📖 335).

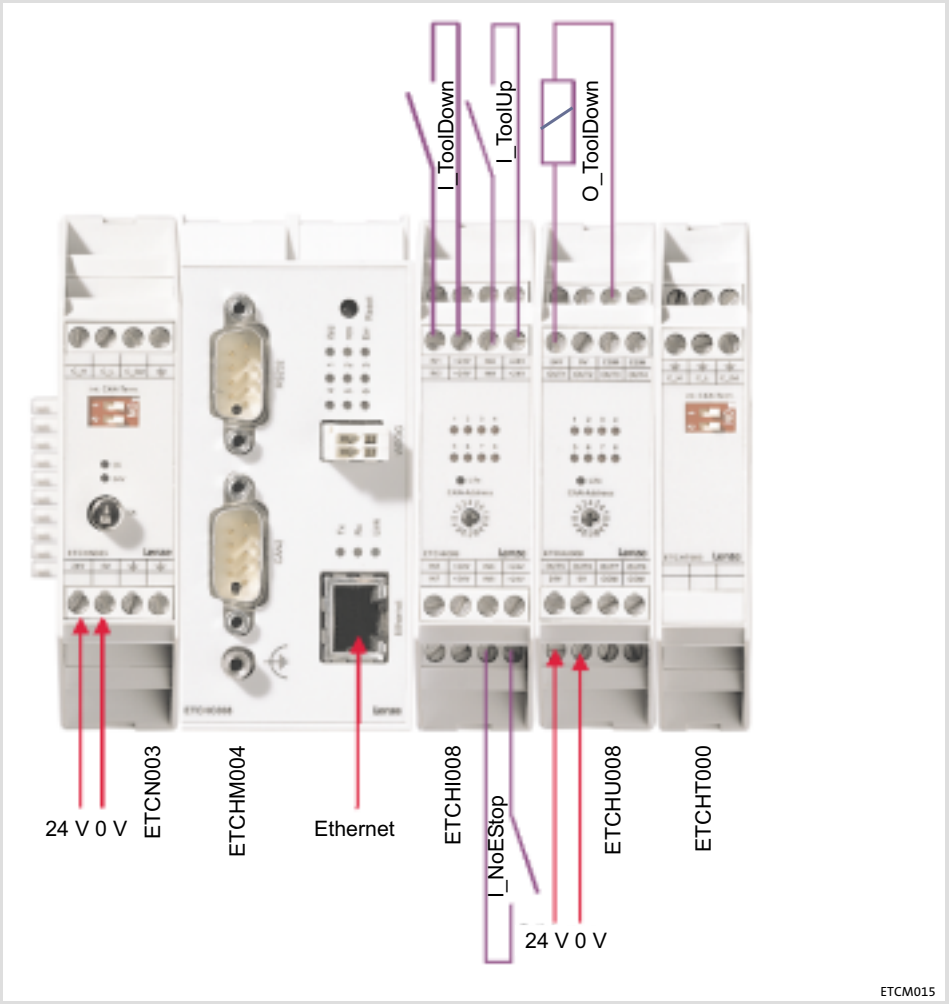


ETCN024

2.12 SPS-Beispielprogramm erstellen

Dieses Kapitel beschreibt die Erstellung eines SPS-Programms am Beispiel eines ETCHx-Systems. Auf die Unterschiede bei einem ETCPx-System wird hingewiesen.

2.12.1 Vorausgesetzte Hardware



- Die 24-V-Spannung ist an den Modulen ETCHN003 (Netzteil) und ETCHU008 (Ausgangsmodul) angeschlossen.
- Ein Schalter **Werkzeug unten** (I_ToolDown) ist am Eingang IN3 des ETCHI008 angeschlossen.
- Ein Schalter **Werkzeug oben** (I_ToolUp) ist am Eingang IN4 des ETCHI008 angeschlossen.
- Ein Schalter **Notaus-Kreis** (I_NoEStop) ist am Eingang IN8 des ETCHI008 angeschlossen.
- Ein Relais **Werkzeug senken** (O_ToolDown) ist am Ausgang OUT1 des ETCHU008 angeschlossen.

2.12.2 SPS-Beispielprogramm starten und konfigurieren

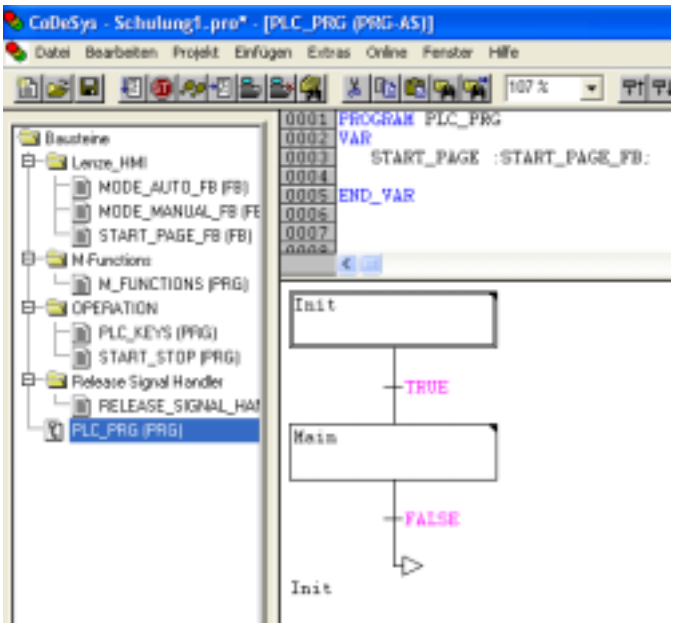
Das Beispielprogramm ”Schulung1” beinhaltet

- ► alle Einstellungen der Steuerung,
- ► die Parametrierung,
- ► die Aufrufe, um das beschriebene CNC-Programm mit den notwendigen SPS-Funktionen für die M-Funktionen zu ergänzen.

Beispielprogramm starten

1. Klicken Sie im Windows-Explorer doppelt auf die Datei ”Schulung1.pro”.

   CoDeSys wird gestartet und das Beispielprogramm wird geladen. Es besteht aus einer Ablaufsteuerung (AS) mit den Schritten ”Init” und ”Main”.

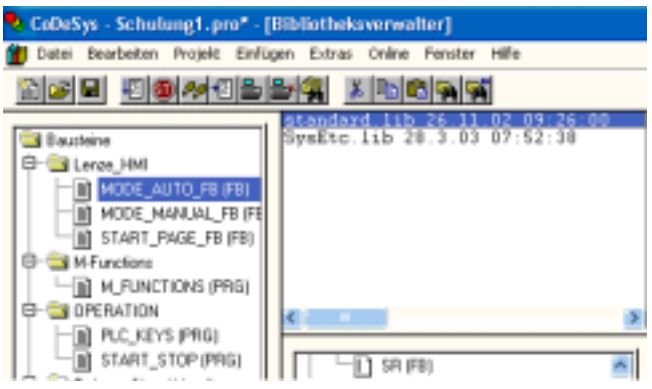


ETCN028

Bibliotheken einfügen

Das Beispielprogramm benötigt die Bibliotheken ”standard.lib” und ”sysetcxc.lib.lib” (...\CoDeSys V2.3\Targets\Lenze\ETCxC).

2. Fügen Sie, falls die Bibliotheken noch nicht vorhanden sind, diese über **Ressourcenverwaltung → Bibliotheksverwalter** ein.

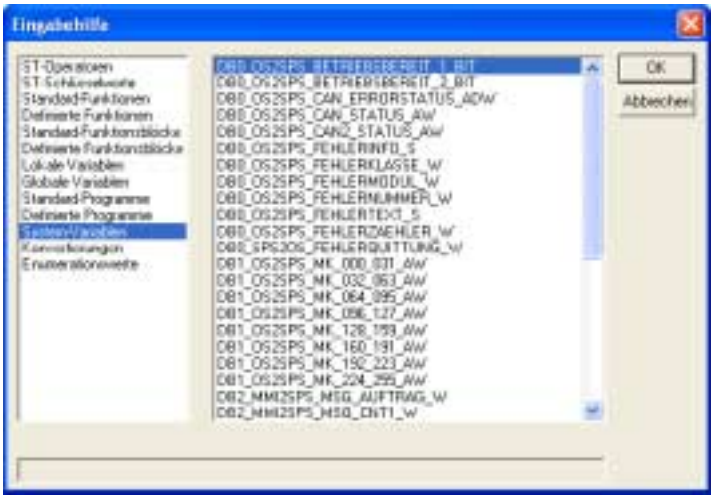


ETCN025

Systemvariablen und Datenbausteine

Die Kommunikation zwischen der CNC-Steuerung und dem SPS-Programm erfolgt über Variablen des Datenbausteins 1, die Kommunikation zwischen dem ETC-MMI und dem SPS-Programm über Variable des Datenbausteins 2. Die unbedingt notwendigen Systemvariablen sind in der folgenden Tabelle zusammengefasst (die Systemvariablen sind in Deutsch und Englisch aufgelistet):

Systemvariable	Merkerwort	Benötigt für
DB1_SPS2NC_NOTAUS_BIT DB1_X0_0_plc2nc_emergency_bit	%MX1.0.0	Freigabe CNC
DB1_SPS2NC_VORSCHUBFREIGABE_W DB1_W1_plc2nc_feed_release_axes_w	%MW1.1	Freigabe CNC
DB1_SPS2NC_VERFAHRTASTENFREIGABE_PLUS_W DB1_W3_plc2nc_release_jog_keys_plus_w	%MW1.3	Tasten Handfahren z. B. Lenze-HMI, bitweise kodiert für 12 Antriebe
DB1_SPS2NC_VERFAHRTASTENFREIGABE_MINUS_W DB1_W4_plc2nc_release_jog_keys_minus_w	%MW1.4	Tasten Handfahren z. B. Lenze-HMI, bitweise kodiert für 12 Antriebe
DB1_SPS2NC_REGLERFREIGABE_W DB1_W5_plc2nc_regulator_release_w	%MW1.5	Freigabe CNC
DB1_SPS2NC_EINLESEFREIGABE_BIT DB1_X7_0_plc2nc_read_in_release_bit	%MX1.7.0	Freigabe CNC
DB1_SPS2NC_PROGRAMMSTART_B DB1_B9_0_plc2nc_program_start_b	%MB1.9.0	ETC-MMI
DB1_SPS2NC_PROGRAMMSTOP_B DB1_B12_0_plc2nc_stop_program_b	%MB1.12.0	ETC-MMI
DB1_NC2SPS_BETRIEBSBEREIT_2_BIT DB1_X128_0_nc2plc_NC_ready_1_bit	%MX1.128.1	Überwachung CNC
DB1_NC2SPS_PROGRAMMSTART_B DB1_B137_0_nc2plc_program_start_b	%MB1.137.0	ETC-MMI
DB1_NC2SPS_PROGRAMMSTOP_B DB1_B142_0_nc2plc_stop_program_b	%MB1.142.0	ETC-MMI



ETCN026

Hinweis!

Die vollständige Liste der Systemvariablen kann im Programmierfenster des ETC-CoDeSys über **<F2>** angezeigt werden.

Aufgabenstellung für M14 und M15

In einem SPS-Programm sollen die beiden Funktionen M14 und M15 wie folgt programmiert werden:

M-Funktion	Soll-Funktion	Programmierung im SPS-Programm
M15	Werkzeug heben	Ausgang ”Werkzeug senken” (O_ToolDown) = 0 Warten auf ”Werkzeug oben” (I_ToolUp) = 1
M14	Werkzeug senken	Ausgang ”Werkzeug senken” (O_ToolDown) = 1 Warten auf ”Werkzeug unten” (I_ToolDown) = 1
M30	Programmende	--

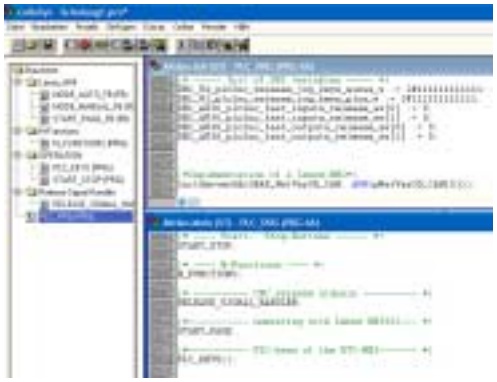
Ablaufsteuerung

Ein Programm in der Sprache ”Ablaufsteuerung” besteht aus den prinzipiellen Elementen ”Schritt”, ”Transition” und ”Verzweigung”. Elemente können in der Ablaufsprache über die rechte Maustaste eingefügt werden.

Das Programm ”Schulung1” besteht aus einem Initialisierungsbaustein mit zwei Schritten und einer Transition.

Element	Name	Inhalt	Kommentar
Schritt	Init	Initialisierung der Systemvariablen	Initialisierungsprogramm
Transition		True	Weiterschaltung auf den nächsten Schritt ”Main”.
Schritt	Main	Baustein START_STOP	Auswertung der Systemvariablen für Start und Stop.
		Baustein M_FUNCTIONS	Bearbeitung der M-Funktionen
		Baustein RELEASE_SIGNAL_HANDLER	Freigabe der Systemvariablen, Auswertung des Eingangs ”Notaus-Kreis” (I_NoEStop).
		Baustein PLC_KEYS	Ausführung der PLC-Keys aus dem ETC-MMI und der M-Funktionen M14 und M15.
Transition		False	Keine Weiterschaltung auf nächsten Schritt, da FALSE. Das Programm bleibt in dem Schritt ”Main”.

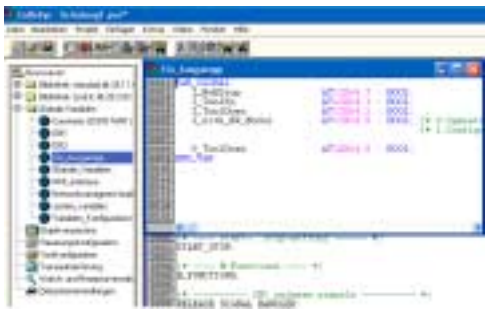
Nach Doppelklick auf einen Schritt öffnet sich ein Fenster, in dem die entsprechende Aktion des Schritts eingetragen ist.



ETCN029

Ein-/Ausgänge definieren

1. Wählen Sie im CoDeSys **Ressourcen → Globale Variablen → Ein_Ausgänge**.
2. Definieren Sie die Ein- und Ausgänge als Bit in den entsprechenden Worten des Prozessabbildes.



ETCN030

Funktion START_STOP

Diese Funktion wertet die folgenden Systemvariablen aus, welche Start und Stopp des SPS-Programms im ETC-MMI steuern:

- DB1_SPS2NC_PROGRAMMSTART_B
- DB1_SPS2NC_PROGRAMMSTOP_B
- DB1_NC2SPS_PROGRAMMSTART_B
- DB1_NC2SPS_PROGRAMMSTOP_B
- (DB1_B9_0_plc2nc_program_start_b,
  DB1_B137_0_nc2plc_program_start_b;
  DB1_B12_0_plc2nc_stop_program_b,
  DB1_B142_0_nc2plc_stop_program_b)

Funktion RELEASE_SIGNAL_HANDLER

Diese Funktion wertet die folgenen Systemvariablen aus, welche die Freigabe- und Überwachungssignale in der Steuerung setzen:

- DB1_SPS2NC_VORSCHUBFREIGABE_W
- DB1_SPS2NC_REGLERFREIGABE_W
- DB1_NC2SPS_BETRIEBSBEREIT_2_BIT
- DB1_W1_plc2nc_feed_release_axes_w
- DB1_W5_plc2nc_regulator_release_w
- DB1_X128_1_nc2plc_NC_ready_2_bit

Funktion M_FUNCTIONS

Der Funktionsbausteins M_FUNCTIONS wertet die von der NC kommenden M-Funktionen aus. Über die beiden folgenden Systemvariablen erkennt der Baustein, ob eine neue M-Funktion beauftragt wurde:

- DB1_NC2SPS_MFKT_STROBE_BIT, DB1_SPS2NC_MFKT_QUITT_BIT
- (DB1_X150_0_nc2plc_M_function_strobe_bit, DB1_X32_0_plc2nc_acknowledgement_for_M_function_bit)

Über die folgende Systemvariable erkennt der Baustein, welche M-Funktion beauftragt wurde:

- DB1_NC2SPS_MFKT_W (DB1_W151_nc2plc_M_function_w)

Solange eine M-Funktion aktiv ist, wird über das Bit DB2_HINT_OPTIONALLYSTOP_ACTIVE_BIT in der Systemvariablen DB2_SPS2HMI_HINWEIS_AW ein Fenster im ETC-MMI geöffnet.

ETCN031

Nach der Erkennung, dass es sich bei der neuen M-Funktion um M14 oder M15 handelt, wird die M-Funktion im Programm PLC_KEYS abgearbeitet.

M14

Der Ausgang *O_ToolDown* wird gesetzt und ein Timer von 5 s gestartet. Wenn der Eingang *I_ToolDown* innerhalb der nächsten 5 s gesetzt wird, wird die Funktion weitergeschaltet, ansonsten erscheint eine Fehlermeldung. Die Fehlermeldung erfolgt über die Funktion SPSERROR, die über eine Fehlernummer (*F_TOOL_UP_KI*) ein Fehlerfenster in dem ETC-MMI öffnet.

```
(*Implementation of the PLC-keys of the ETC-HMI *)
(*F1: DRIVES ON/OFF (in release signal handler) *)
(*F4: Tool open/close                           *)

(* tool down *)
SR_Tool_down(SET1:=MMI_PLC_key_tool_down);

TRIG_M14(CLK:=M_M14_bit OR SR_Tool_down.Q1);
IF (M_M14_bit OR SR_Tool_down.Q1) THEN
   IF (TRIG_M14.Q) THEN
      TIMER_M14(IN:=FALSE);
      SR_Tool_down.RESET:=FALSE;
      O_ToolDown := TRUE;
   ELSE
      TIMER_M14(IN:=TRUE, PT:=#5s);
      IF (I_ToolDown OR Tool_down_q) THEN
         M_M14_bit := FALSE;
         SR_Tool_down.RESET:=TRUE;
      ELSIF (TIMER_M14.Q) THEN
         M_M14_bit := FALSE;
         SR_Tool_down.RESET:=TRUE;
         SPSERROR(F_TOOL_UP_KI, 3, '', 0);
      END_IF
   END_IF
END_IF
```

ETCN032

M15

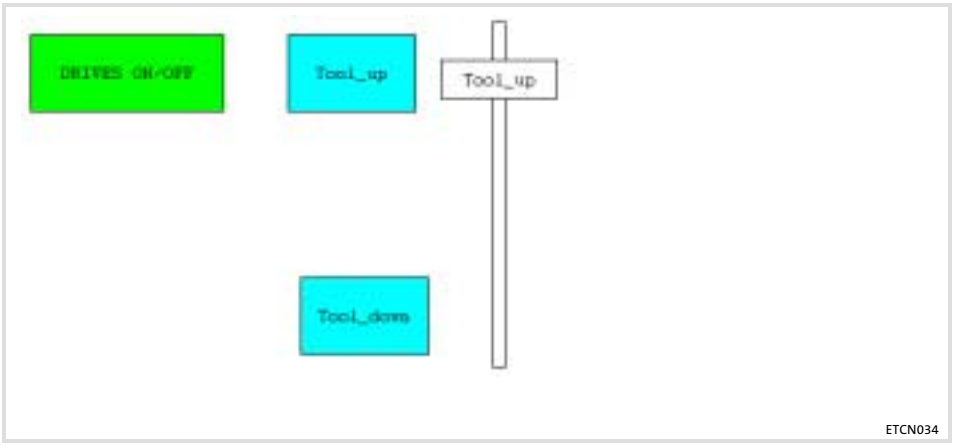
Der Ausgang *O_ToolDown* wird zurückgesetzt und ein Timer von 5 s gestartet. Wenn der Eingang *I_ToolUp* innerhalb der nächsten 5 s gesetzt ist, wird die Funktion weitergeschaltet, andernfalls erscheint eine Fehlermeldung.

```
(* tool up *)
SR_Tool_up(SET1:=MMI_PLC_key_tool_up);

TRIG_M15(CLK:=M_M15_bit OR SR_Tool_up.Q1);
IF (M_M15_bit OR SR_Tool_up.Q1) THEN
   IF (TRIG_M15.Q) THEN
      TIMER_M15(IN:=FALSE);
      SR_Tool_up.RESET:=FALSE;
      O_ToolDown := FALSE;
   ELSE
      TIMER_M15(IN:=TRUE, PT:=t#5s);
      IF (I_ToolUp OR Tool_up_g) THEN
         M_M15_bit := FALSE;
         SR_Tool_up.RESET:=TRUE;
      ELSIF (TIMER_M15.Q) THEN
         M_M15_bit := FALSE;
         SR_Tool_up.RESET:=TRUE;
         [illegible](P_TOOT_[illegible], 3, '', 0);
      END_IF
   END_IF
END_IF;
```

ETCN033

Sind an der Steuerung keine E/A-Module angeschlossen, kann die Visualisierung genutzt werden.



ETCN034

Mit ”DRIVES ON/OFF” werden die Antriebe ein- und ausgeschaltet. ”Tool_up” und ”Tool_down” stellen die Schalter dar, die betätigt werden sollen.

2.12.3 SPS-Beispielprogramm in die Steuerung laden

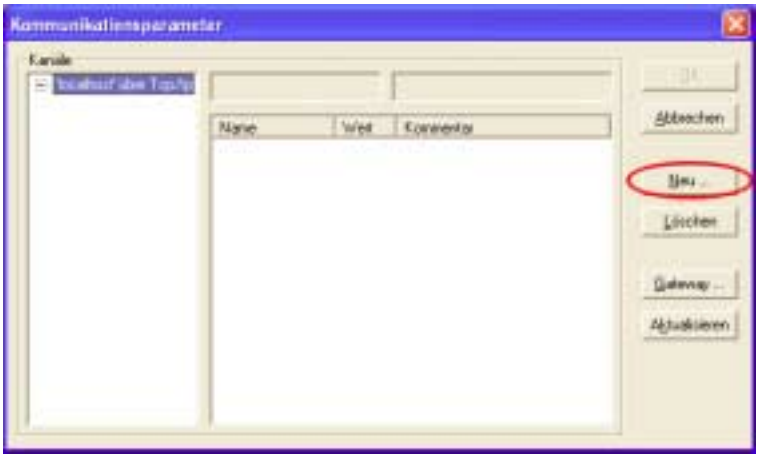
Hinweis!

Die in diesem Kapitel beschriebenen Schritte gelten nur für die ETCHx-Variante (Hutschienen-Variante); für die ETCPx-Variante (PCI-Karte) sind sie nicht erforderlich.

Über das ETC-CoDeSys kann die Verbindung vom PC zur Steuerung eingestellt werden, wahlweise über eine serielle Verbindung (RS232-Schnittstelle) oder über das Netzwerk.

Serielle Verbindung

1. Verbinden Sie PC und ETC über die serielle Schnittstelle.
2. Wählen Sie **Online → Kommunikationsparameter** .
3. Klicken Sie auf **Neu**.

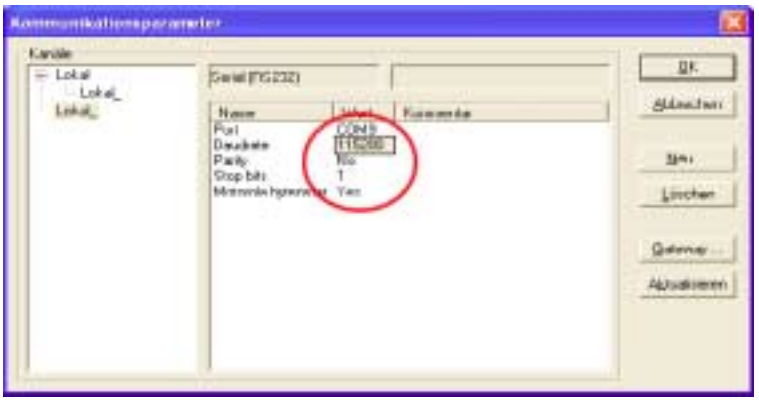


ETCM036

4. Geben Sie einen Verbindungsnamen ein (z. B. lokal_) und wählen Sie ”Serial RS232”.

ETCM037

5. Klicken Sie auf **OK**.
6. Geben Sie die Parameter der Schnittstelle (z. B. COM1 mit Baudrate, Parity und Stopbit) vor, klicken Sie dazu auf das entsprechende Feld doppelt.

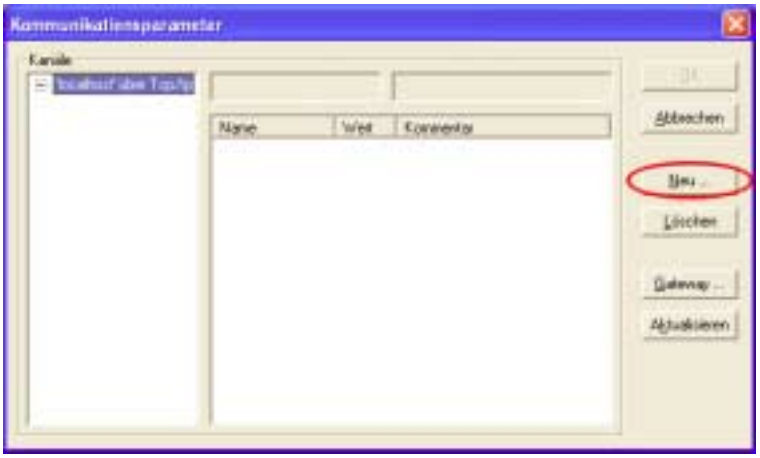


ETCM038

7. Da die Steuerung einen Motorola-Prozessor beinhaltet, muss das Feld ”Motorola Byteorder” auf ”Yes” stehen. Korrigieren Sie das ggf.
8. Schließen Sie das Fenster mit **OK**.

Verbindung über Netzwerk

1. Wählen Sie **Online → Kommunikationsparameter**.
2. Klicken Sie auf **Neu**.

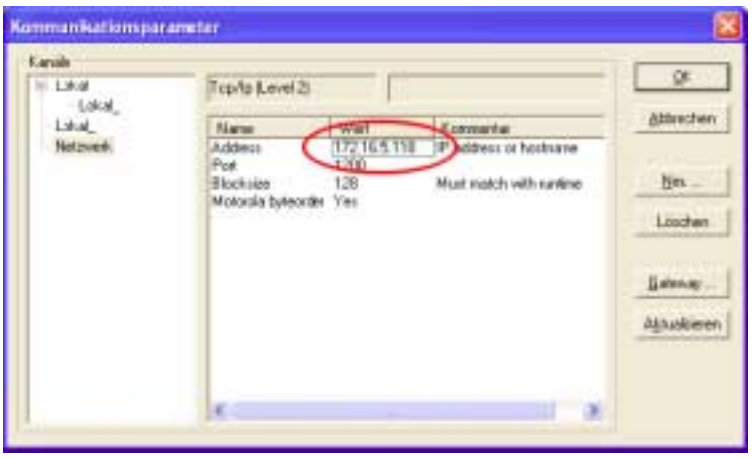


ETCM036

3. Geben Sie einen Namen vor (z. B. Netzwerk) und wählen Sie als Parameter das TCP/IP-Protokoll. Klicken Sie auf **OK**.

ETCM040

4. Klicken Sie doppelt auf den Wert des Feldes ”Adress” und geben Sie die korrekte IP-Adresse vor (📖 28).



ETCM041

Hinweis!

Verwenden Sie in IP-Adressen keine führenden Nullen, da die IP-Adresse sonst als Oktadezimalzahl interpretiert wird.

5. Da die Steuerung einen Motorola-Prozessor beinhaltet, muss das Feld ”Motorola Byteorder” auf ”Yes” stehen. Korrigieren Sie das ggf. durch mehrfaches Doppelklicken.

Hinweis!

Im linken Fensterbereich werden die Verbindungsmöglichkeiten angezeigt. In diesem Fall die beiden Verbindungen ”Lokal_” und ”Netzwerk”.

Die aktuelle Einstellung ist die Einstellung, die vor Klicken auf **OK** im rechten Fenster angezeigt wird; sie wird für das Einloggen benutzt.

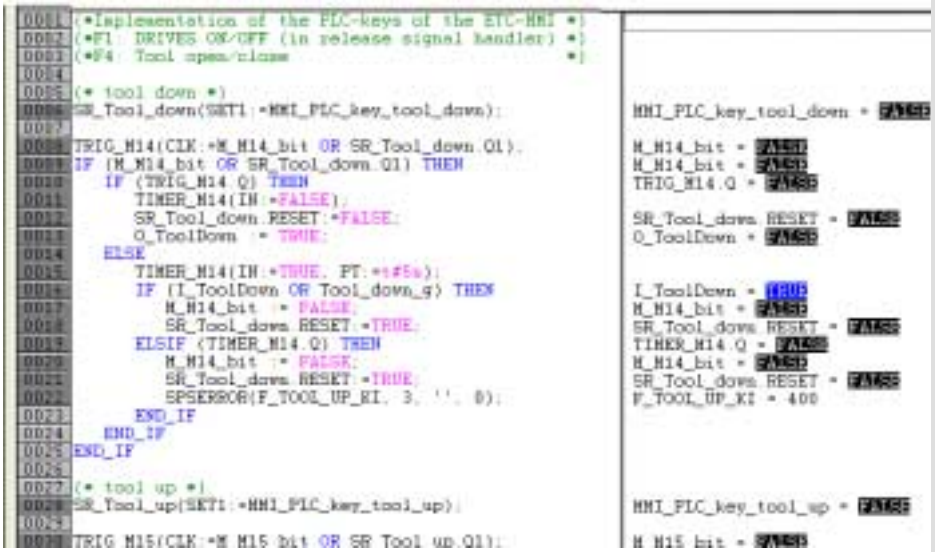
6. Schließen Sie das Fenster mit **OK**.

Programm laden und starten

1. Wählen Sie **Online → Einloggen**.
   Das Programm wird in die Steuerung geladen.
2. Wählen Sie **Online → Start**.
   Das Programm wird gestartet. Nach erfolgreichem Start wechselt in der Statuszeile das Feld ”LÄUFT” seine Farbe von grau auf schwarz.

ETCM042

In der Statuszeile wird die aktuelle Verbindung (hier: ”Lokal_”) angezeigt. Das Programm zeigt im rechten Teil des Fensters die aktuellen Werte der Variablen.



ETCNC097

Programm bedienen

Nach Einloggen in die Steuerung über **Online → Einloggen**, können folgende Bedienfunktionen aktiviert werden:

- ► Programm stoppen: **Online → Stop**
- ► Datenspeicher auf Startwerte zurücksetzen: **Online → Reset**
- ► Boot-Projekt erstellen: **Online → Bootprojekt erzeugen**
  Das Programm wird als ”default.prg” in das Flash der Steuerung abgelegt. Beim Starten der Steuerung wird nach einem Programm ”default.prg” gesucht. Dieses wird dann automatisch als Boot-Projekt gestartet.
- ► Boot-Projekt löschen: **Online → Reset (Ursprung)**
- ► Kommunikation mit der Steuerung beenden: **Online → Ausloggen**
  Das Programm in der Steuerung behält den Status, der zuletzt ausgewählt wurde.

2.13 CNC- und SPS-Programm testen

Hinweis!

Der Betrieb des CNC-Programms ist grundsätzlich möglich

- ohne angeschlossene Mechanik und Antriebe.
  Dies wird erreicht durch Setzen der Maschinenkonstanten MK_TEST_OHNEMECHANIK=1.
- ohne ein SPS-Programm.
  Dies wird erreicht durch Setzen der Maschinenkonstanten MK_SPS_DUMMY=1.

Für die folgenden Tests mit Programm und Mechanik müssen daher die Maschinenkonstanten wie folgt eingestellt werden:

Testeinstellung

MK-Schlüsselwort	Anzahl Werte	Werte
MK_TEST_OHNEMECHANIK	1	0
MK_SPS_DUMMY	1	0

Hardware beschalten

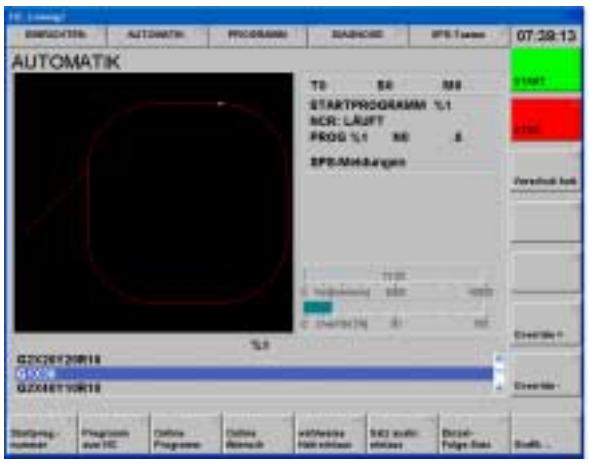
1. Beschalten Sie den Hardware-Aufbau wie vorausgesetzt (📖 58).
   – Die Eingänge ”Werkzeug unten” und ”Werkzeug oben” müssen entsprechend dem Fortschritt des Programms (M14, M15) gesetzt werden. Die Eingänge müssen daher fest auf 24 V gelegt werden, oder rechtzeitig geschaltet werden.

SPS-Programm starten

1. Starten Sie das SPS-Beispielprogramm im CoDeSys. Klicken Sie dazu im Windows-Explorer doppelt auf die Datei ”Schulung1.pro” (📖 59).
   CoDeSys wird gestartet und das Beispielprogramm wird geladen.
2. Laden Sie das SPS-Programm in die Steuerung über **Online → Einloggen** (📖 66).
3. Starten Sie das SPS-Programm über **Online → Start** (📖 66).

CNC-Programm starten

1. Starten Sie das ETC-MMI über **<Start> → Programme → Lenze → ETC** und drücken Sie im ETC-MMI-Fenster **<F11>** (Programm).
   Das MMI-Fenster der Betriebsart ”Programmieren” wird geöffnet. Es zeigt den Texteditor für das CNC-Programm.
2. Drücken Sie **<F8>** (Grafik) und anschließend **<F7>** (Grafik ein/aus).
   Das Fenster wird geteilt. Es zeigt links den Texteditor und rechts ein Anzeigefeld für die grafische Darstellung (Grafikbereich).
3. Drücken Sie **<F2>** (Programm öffnen).
4. Wählen Sie das CNC-Programm (z. B. **test.din**) und drücken Sie **<Enter>**.
5. Drücken Sie **<F7>** (Programm zum NC).
6. Wählen Sie das CNC-Programm (z. B. **test.din**) und drücken Sie **<Enter>**.
   Das CNC-Programm wird auf die Steuerung übertragen (📖 49).
7. Drücken Sie **<S1>** (Start).
   Die Antriebe starten und die Kontur des Rechtecks wird abgefahren (📖 49).
   Die gezeichnete Gerade vom Nullpunkt (0/0) zum Anfangspunkt (10/10) wird nur dann abgefahren, wenn der Antrieb auch im Nullpunkt steht. Andernfalls fährt die Steuerung die Antriebe auf einer Geraden vom aktuellen Punkt zum Anfangspunkt.
8. Ändern Sie die Bahngeschwindigkeit auf einen kleinen Wert (z. B. 1 %), klicken Sie dazu auf **<S7>** (Override -).
   Bei korrektem Laden des SPS- und CNC-Programms läuft eine Marke die kreisförmige Kontur ab. Der Ausgang ”Werkzeug senken” wird entsprechend dem aufgerufenen M-Funktion gesetzt und zurückgesetzt.



ETCN035

9. Um die Antriebe zu stoppen, drücken Sie **<S2>** (Stop).

Fehlermeldungen des CNC-Programms

Wenn einer der beiden Eingänge nicht rechtzeitig auf 24 V liegt, stoppt das Programm und die folgende Fehlermeldung erscheint in einem roten Kasten.

ETCN036

Diese Fehlermeldung wird über die Funktion **SPSERROR** im Modul **PLC_KEYS** nach Ablauf eines Timers mit 5 s erzeugt. Der zu der Fehlernummer zugehörige Text ist in der Datei ”sps_fehl.db” (englische Version: ”sps_erro.db”) eingetragen. Diese Datei liegt in dem Verzeichnis ”cfg” des ETC-MMI.

Im Bild handelt es sich um einen Fehler aus der Steuerung (2) des Moduls SPS-Programm (10) mit der Fehlernummer 400 (F_TOOL_UP_KI). Für eine korrekte Fehlermeldung muss daher in der Datei ”sps_fehl.db” eine Fehlernummer 2.10.400 mit entsprechendem Text eingetragen werden.

2.14 SPS-Tasten im ETC-MMI

In der Betriebsart ”SPS-TASTEN” ist es möglich, Handfunktionen in der SPS vom ETC-MMI aus zu starten. Hierzu stehen 6 Menüebenen zur Verfügung. Das entspricht 42 Funktionstasten.

In diesem Beispiel sollen drei SPS-Tasten wie folgt belegt werden:

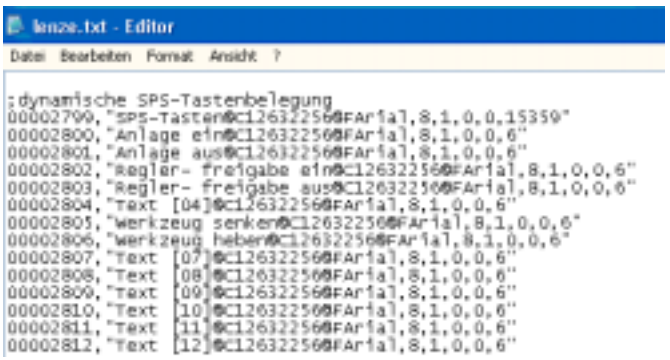
F1: Anlage ein/aus

F4: Werkzeug senken

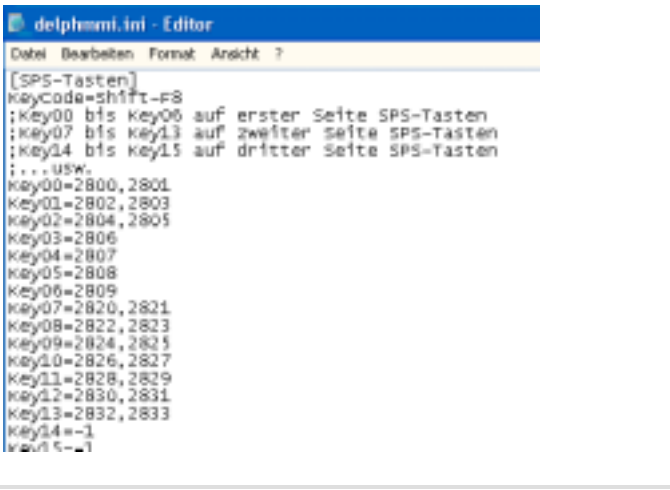
F5: Werkzeug heben

2.14.1 Beschriftung der SPS-Tasten im ETC-MMI

Die Beschriftung der SPS-TASTEN wird in der sprachabhängigen **lenze.txt**-Datei durchgeführt.



Diese Texte werden dann durch die Konfigurierungsdatei **delphmmi.ini** den Tasten zugewiesen:



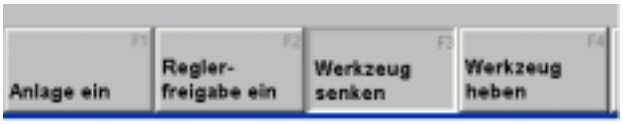
Key 00: Für die erste SPS-Taste gelten die Texte 2800 und 2801 aus der Datei lenze.txt.

2 Texte für eine Taste beinhalten automatisch eine Tooglefunktion. D. h. mit Betätigen von F1 wird die Anlage eingeschaltet und beim erneuten Betätigen ausgeschaltet. Der Ausgabewert in der SPS wird durch jedes Betätigen wieder umgeschaltet.

Key 03: Hier gilt Text 2805 (Werkzeug senken) der Datei ”lenze.txt”.

Key 04: Hier gilt Text 2806 (Werkzeug heben) der Datei ”lenze.txt”.

Darstellung im ETC-MMI:



ETCN039

2.14.2 Aufruf der Signale in der SPS

ETCN040

Die Zuweisung geschieht im Beispiel absolut auf die Wortvariable:

MMI_PLC_key_DRIVES_ON → %MX1.121.0 (X=Bit) greift auf Bit 0 des DW212 im DB1 zu.

Ein Zugriff auf die symbolische Systemvariable ist ebenfalls möglich:

DB1_AW212_mmi2plc_jog_key_aw [0].0

DB2_MMI2SPS_TASTEN_AW [0].0

2.14.3 Funktion der Tasten

Die Tasten im ETC-MMI haben die gleiche Funktion wie die M-Funktionen M14 und M15; d. h. mit den Tasten kann das Werkzeug gesenkt und gehoben werden. Auch hier muss der Eingang bzw. das Signal der Visualisierung innerhalb von 5 s anliegen, da andernfalls eine Fehlermeldung erscheint.

2.15 Bedienung über ein Lenze-HMI

Neben der Bedienung über das ETC-MMI (mit PC), ist eine Bedienung über ein Lenze-HMI möglich.

Das über den CAN1-Bus verbundene HMI kann als Zusatz- oder als Hauptbedienteil eingesetzt werden.

Wird das HMI zur kompletten Maschinensteuerung eingesetzt, muss die Steuerung vorher mit einem PC parametriert und programmiert werden.

In diesem Beispiel wird das HMI505 als Zusatzbedienteil neben dem ETC-MMI eingesetzt.

Neben der Funktion des Handfahrens kann vom HMI505 das NC-Programm %1 gestartet werden. Anhand eines einfachen HMI-Projektes wird diese Funktion erläutert.

Das HMI505-Projekt liegt als HMI-Designer Datei **Schulung1_HMI505_V1.VTS** auf der ETC-CODeSys-CD vor.

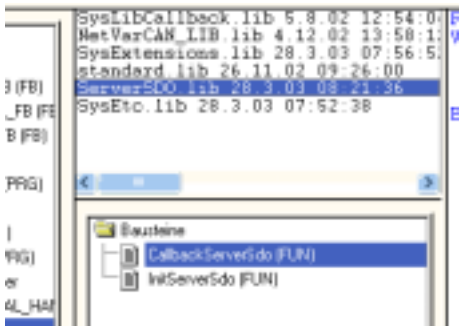
2.15.1 Einstellungen für den Anschluss eines Lenze-HMI H505

Bibliotheken einbinden

Aktivierung der Kommunikationsfunktion im Initialisierungsschritt:

ETCN041

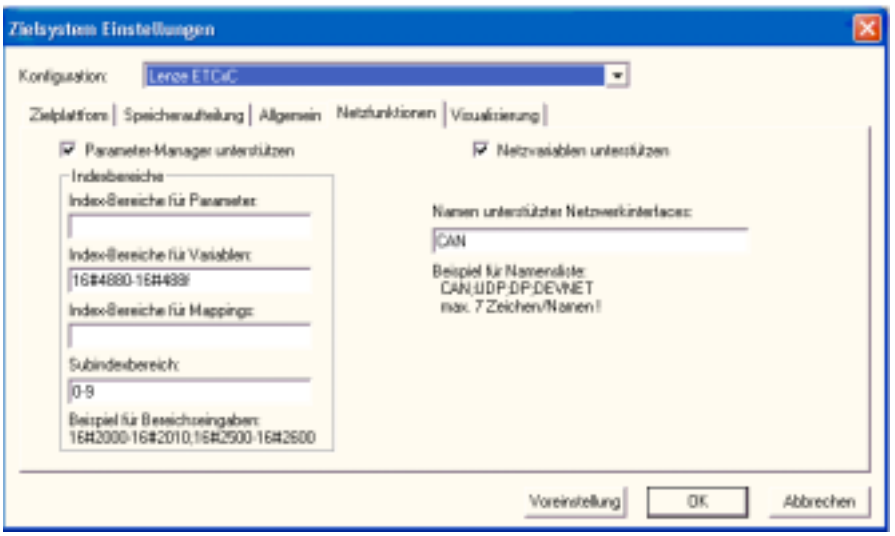
Um diese Funktion zu nutzen, muss die Bibliothek ”ServerSDO.lib” eingebunden werden. Dabei werden noch weitere Bibliotheken eingebunden. Diese Bibliotheken stellen die Schnittstellenfunktionen von der ETC zum HMI zur Verfügung:



ETCN042

Einstellungen im SPS-Programm (Zielsystem)

Um der Steuerung mitzuteilen, dass eine Kommunikation zu einem HMI gewünscht wird, ändern Sie die Zielsystemeinstellung wie folgt :



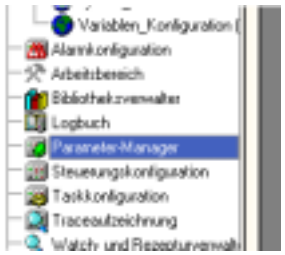
ETCN043

In den Zielsystemeinstellungen sind auf der Reiterkarte ”Netzfunktionen”, das Objektverzeichnis und die Netzwerkvariablen zu aktivieren.

Das Objektverzeichnis stellt dem HMI die deklarierten Variablen mit den dazugehörigen Indices zur Verfügung (Lenze-Codestelle = Index bei CANopen).

Die gewünschte Anzahl von Indices kann im Indexbereich eingestellt werden. Als Netzwerkinterface zum HMI fungiert das CAN-Netzwerk.

Nachdem diese Einstellungen vorgenommen worden sind, stellt das CoDeSys das Objektverzeichnis zur Verfügung:



ETCN044

CoDeSys-Objektverzeichnis

Das Objektverzeichnis bildet die Schnittstelle zwischen den Codestellen im HMI H505 und den HEX-Indices in der ETC. Nach Lenze-Norm gibt es diesen Zusammenhang:

INDEX =DEZ_TO_HEX(24575-Codestelle)

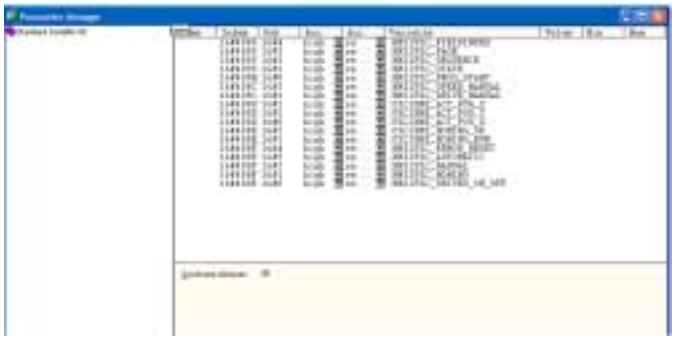
Um diese Umrechnung zu erleichtern, steht eine Excel-Tabelle zur Verfügung (Umrechnung Objektverzeichnis Schulung.xls). Diese befindet sich auf der ETC-CODeSys-CD im Verzeichnis "Systemhandbuch". In der Datei kann die Deklaration und die Aufteilung der HMI-Variablen auf die Codestellen vorgenommen werden.

Lenze Codestelle	Sub-code	HEX-Index	HEX Index	Sub-Index	Zugriff	Attribut	Variable
6010	4	4885	16V4885	4	high	read-write	.HMI2PLC_FIELDINDEX
6010	3	4885	16V4885	3	high	read-write	.HMI2PLC_PAGE
6010	2	4885	16V4885	2	high	read-write	.HMI2PLC_SEQUENCE
6010	1	4885	16V4885	1	high	read-write	.HMI2PLC_STATE
6004	0	488B	16V488B	0	high	read-write	.HMI2PLC_PROG_START
6003	3	488C	16V488C	3	high	read-write	.HMI2PLC_SPEED_MANUAL
6003	1	488C	16V488C	1	high	read-write	.HMI2PLC_DRIVE_MANUAL
6002	3	488D	16V488D	3	high	read-write	.PLC2HMI_ACT_POS_Z
6002	2	488D	16V488D	2	high	read-write	.PLC2HMI_ACT_POS_Y
6002	0	488D	16V488D	0	high	read-write	.PLC2HMI_ACT_POS_X
6001	2	488E	16V488E	2	high	read-write	.PLC2HMI_HOMING_OK
6001	1	488E	16V488E	1	high	read-write	.PLC2HMI_HOMING_RUN
6000	4	488F	16V488F	4	high	read-write	.HMI2PLC_ERROR_RESET
6000	3	488F	16V488F	3	high	read-write	.HMI2PLC_AUTOMATIC
6000	2	488F	16V488F	2	high	read-write	.HMI2PLC_MANUAL
6000	1	488F	16V488F	1	high	read-write	.HMI2PLC_HOMING
6000	0	488F	16V488F	0	high	read-write	.HMI2PLC_DRIVES_ON_OFF

ETCN045

Der gelb markierte Teil kann komplett mit **<Strg>+<C>** in das Objektverzeichnis von CoDeSys kopiert werden.

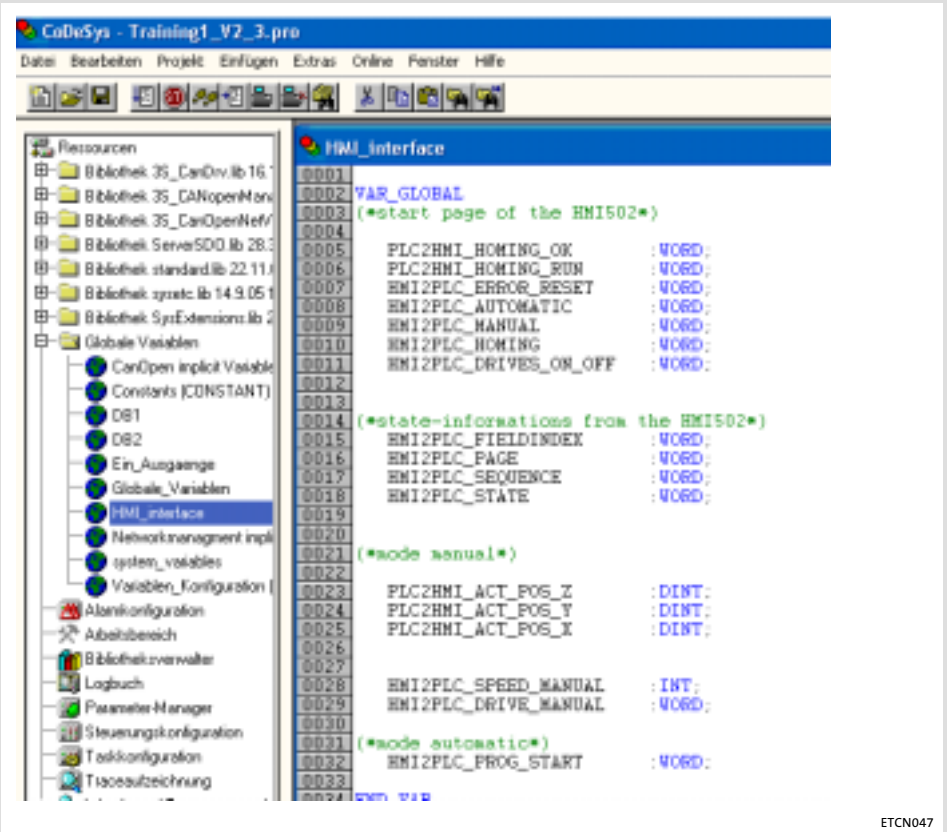
Objektverzeichnis (Reiterkarte Variable):



ETCN046

Achten Sie besonders auf den Punkt vor der Variablen.

Die deklarierten Variablen müssen als Globale Variablen angelegt werden:



Greift das HMI auf Variablen zu, die im CoDeSys nicht angelegt sind, kommt es zu einem Kommunikationsfehler.

2.15.2 Funktionsbeschreibung HMI505-Bedienung

Im Folgenden wird kurz auf die Funktionsbeschreibung des Projekts ”ETC-Schulung des HMI” eingegangen.

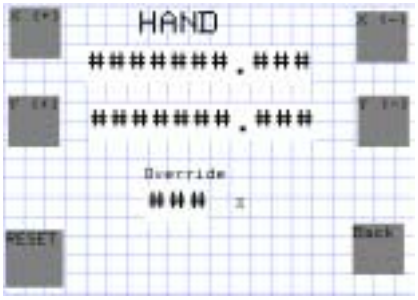
Startseite

- ► Anwahl der Betriebsarten Hand und Auto
- ► Ein-, Ausschalten der Antriebe
- ► Fehlerreset
- ► Aktivierung der Downloadseite

ETCN048

Betriebsart Hand

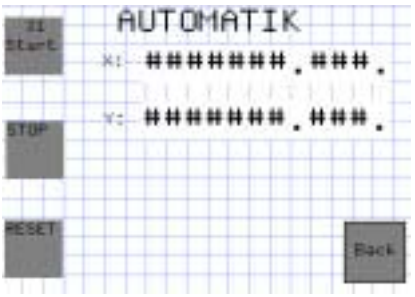
- ► Einstellen des Overrides für das Handverfahren der Achsen
- ► Verfahren der beiden Achsen X und Y in beide Richtungen
- ► Fehlerreset
- ► Rückspung zur Startseite
- ► Anzeige der Achspositionen



ETCN049

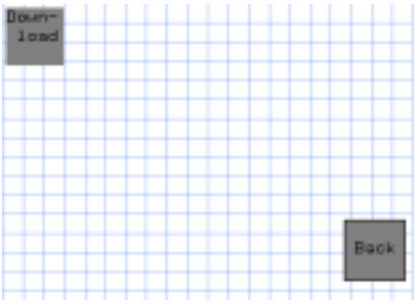
Betriebsart Auto

- ▶ Fehlerreset
- ▶ Rücksprung zur Startseite
- ▶ Anzeige der Achspositionen
- ▶ Start des NC-Programms %1
- ▶ Stop des laufenden NC-Programms



ETCN056

Downloadseite



ETCN057

Projektname: **Schulung1_HMI505_V1.VTS**

2.16 Firmware der ETCHx in der Betriebsart ”Stand-Alone” aktualisieren

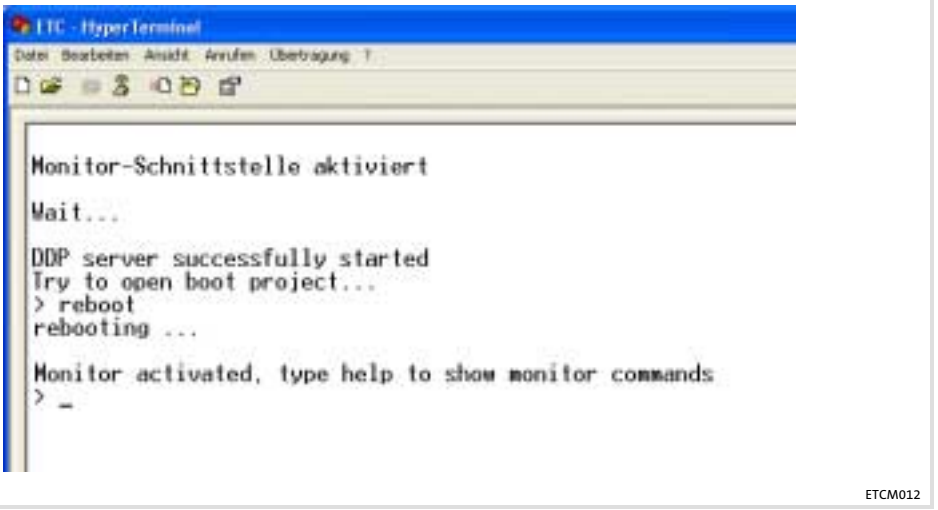
Hinweis!

Die in diesem Kapitel beschriebenen Schritte gelten nur für die ETCHx-Variante (Hutschienen-Variante); für die ETCPx-Variante (PCI-Karte) sind sie nicht erforderlich.

2.16.1 Aufruf des Bootmonitors in der Steuerung

Ausgangszustand: Die Kommunikation zwischen PC und ETCHx ist aufgebaut und die Monitorschnittstelle ist aktiviert (📖 21)

1. Geben Sie im Terminal-Programm den Befehl **reboot** ein.
2. Halten Sie während des laufenden Reboot-Prozess die Tasten **<Shift> + <!>** gedrückt.



Wenn der Prompt ”>” erscheint, ist die Eingabe im Bootmonitor der Steuerung aktiv. Auf der Steuerung blinken die LEDs 1 ... 6 umlaufend.

Hinweis!

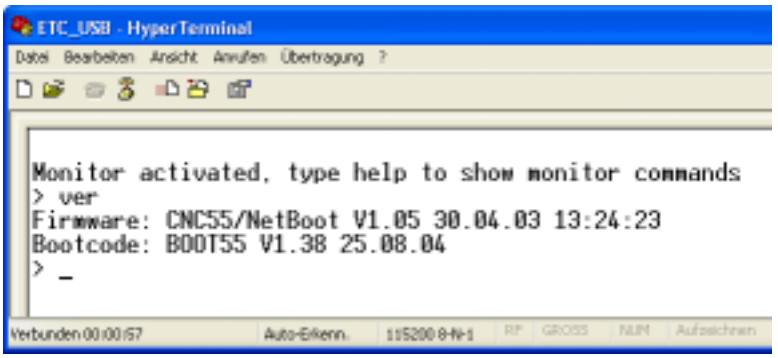
Der Reset der Steuerung kann bei laufender Firmware auch durch Aus-/Einschalten der Steuerung oder Drücken des Resetknopfs erfolgen.

2.16.2 Versionsabfrage der Firmware

1. Wenn der Prompt ”>” des Bootmonitors erscheint, geben Sie den Befehl **ver** ein.

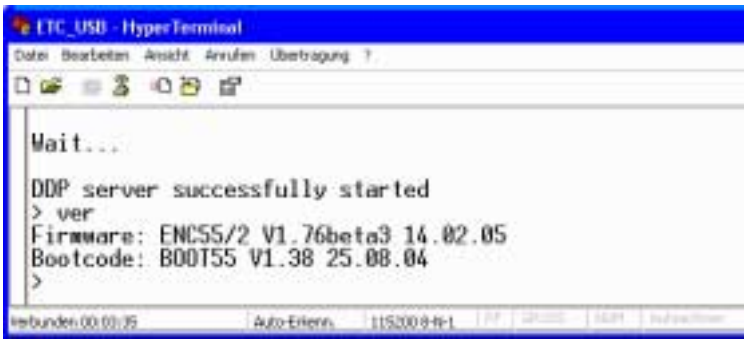
   Die Version der Firmware wird abgefragt und angezeigt.

Variante ”mit ETC-MMI”



ETCN005

Variante ”Stand-Alone”

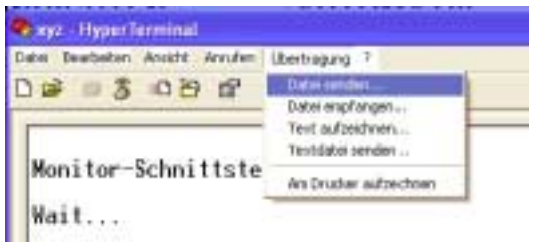


ETCN006

2.16.3 Updaten der Firmware

Ausgangszustand: Die Kommunikation zwischen PC und ETC ist aufgebaut und die Monitorschnittstelle ist aktiviert (📖 21).

1. Um die aktuelle Firmware auf die ETC zu übertragen, aktivieren Sie im HyperTerminal den Menüpunkt **Übertragung → Datei senden**.



ETCM011

2. Geben Sie den Pfad zur Firmware-Datei vor und wählen Sie als Protokoll ”Zmodem mit Wiederherstellung”.

   Firmware-Dateinamen: ETCHC_A.rsc (ETCHC mit 4 MB) oder ETCPC.rsc (ETCPC)

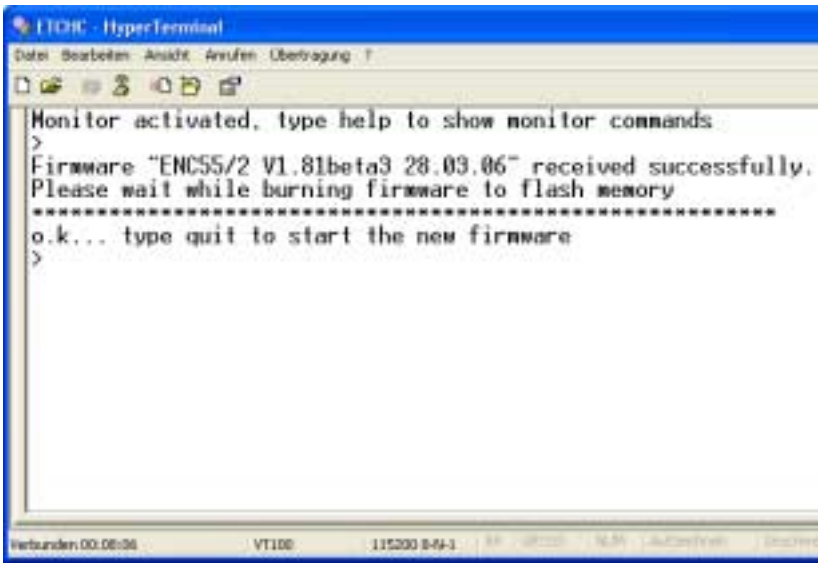
ETCN098

3. Klicken Sie auf **Senden**.

   Während des Ladens der Datei in die Steuerung erscheint das nachfolgende Bild. Auf der ETC-Frontplatte blinken die LEDs 1 ... 6 umlaufend.

ETCN099

   Nach Abschluss des Ladevorgangs erscheint das folgende Bild:


ETCHC - HyperTerminal
Datei Bearbeiten Ansicht Anrufen Übertragung ?
Monitor activated, type help to show monitor commands
>
Firmware "ENC55/2 V1.81beta3 28.03.06" received successfully.
Please wait while burning firmware to flash memory
****
o.k... type quit to start the new firmware
>
VT100

ETCN100

   Die Steuerung wartet auf das Laden der Firmware; auf der Frontplatte der ETC blinken die LEDs 1 ... 6 umlaufend.

4. Geben Sie imHyperTerminal-Fenster **quit** ein.

   Die Firmware startet. Auf der ETC-Frontplatte leuchtet die grüne Watchdog-LED.

Wichtige Befehle des Bootmonitors

Befehl	Bedeutung
ver	Zeigt die Versionen des Bootloaders und der aktuell geladenen Firmware an.
sz [dateiname]	Sendet die aktuell geladene Firmware unter dem angegebenen Namen per Z-Modem an den PC.
rz	Dieser Befehl wird implizit durch das angeschlossene Terminalprogramm bei Anwahl von ”Dateien Senden” an die Steuerung geschickt und dient zur Übertragung von Firmware oder Bootloader in die Steuerung. Zum Senden muss das Protokoll ”ZMODEM” angewählt werden.
quit	Beendet den Bootmonitor und startet die geladene Firmware.
reset	Setzt die Steuerung zurück.
help	Zeigt die Befehle des Bootmonitors an.

3 CNC-Programmierung

Dieses Kapitel stellt die Funktionen der Steuerung ETCxC dar, mit denen die Programme zur Bearbeitung von Werkstücken erstellt werden. Die Programmierung der ETCxC erfolgt in Anlehnung an die DIN 66025. Der Befehlsvorrat ist gegenüber der DIN mit einer Vielzahl zusätzlicher, leistungsfähiger Funktionen erweitert.

Im Folgenden werden zunächst einige grundlegende Bedingungen für den Aufbau eines Programms beschreiben. Anschließend werden alle Funktionen im Detail beschrieben.

3.1 Grundlagen

Varianten der Programm-Erstellung

Die Erstellung der Programme kann entweder an der Steuerung selbst vorgenommen werden (📖 47) oder extern, mit einem beliebigen ASCII-Editor entsprechend nachfolgender Regeln.

Programm übertragen

Wenn die Steuerung über Ethernet-Schnittstelle an ein ETC-MMI angeschlossen ist, werden die Programme beim Start der Steuerung von der Festplatte des PCs direkt in die Steuerung übertragen.

Falls kein ETC-MMI als Bediengerät verwendet wird, können die Programme mittels Modem über die Monitorschnittstelle (serielle Schnittstelle) der Steuerung übertragen werden. In der Steuerung gespeicherte Programme können dann auch über die Monitorschnittstelle ausgelesen werden.

Um Programme über die Monitorschnittstelle oder eine Diskette in die Steuerung zu übertragen, müssen diese als Namen eine bis zu 4-stellige Nummer und die Extension **.DIN** haben. Die Nummer wird dabei automatisch als Programmnummer verwendet und mit einem vorangestellten %-Zeichen am Anfang des Programms eingefügt. Die Programme werden automatisch im EEPROM der Steuerung gespeichert, falls die Maschinenkonstante MK_NCPROG_NICHT_INS_EEPROM = 0 ist.

Bei Verwendung eines ETC-MMI können auch Online-Programme abgearbeitet werden. Solche Programme können beliebig lang sein und werden während der Übertragung abgearbeitet. Online-Programme müssen sequenziell aufgebaut sein und dürfen keine Sprünge oder andere Verzweigungen enthalten. Unterprogrammaufrufe mit ”G22 Lxxx” sind erlaubt.

Programm-Komponenten

Ein Programm besteht aus einer Programmnummer (Programmanfang), einer Anzahl von Sätzen und dem Programmende. Es muss mindestens aus Programmanfang und Programmende bestehen. Ein Programm darf beliebig viele Sätze umfassen. Begrenzt wird es lediglich durch den zur Verfügung stehenden Gesamtspeicher. Die Anzahl der Programme ist auf einen einstellbaren Wert begrenzt (📖 198).

In einem Programm dürfen keine Steuerzeichen verwendet werden.

Programmanfang

Ein Programm beginnt in der Regel mit einem ”%”-Zeichen, dem die Programmnummer (1 ... 9999) folgt.

Beim Laden eines Programms über die Monitorschnittstelle wird diese Programmnummer automatisch aus dem Dateinamen gebildet und am Anfang des Programms eingefügt (123.DIN -> %123).

Jedes Programm kann direkt gestartet oder als Unterprogramm durch andere Programme aufgerufen werden. Dabei ist im Einzelfall die Bereitstellung von Parametern zu beachten.

Eine Unterscheidung von Haupt- und Unterprogrammen anhand der Programmnummer existiert nicht.

Programmnummern ≥ 8000 werden im Allgemeinen gesondert betrachtet. In diesem Bereich werden normalerweise Zyklenprogramme abgelegt.

Satz

Ein Satz besteht aus einer Anweisung, der optional eine Satznummer vorangestellt werden kann (siehe unten). Eine Anweisung setzt sich aus folgenden Elementen zusammen:

- ► Wegbedingung ”G”
- ► Hilfsfunktion ”M”, ”S”, ”T”
- ► Schnelle Ausgänge ”Q”
- ► Parameterzuweisungen
- ► Rechenanweisung
- ► Kommentar

Alle Elemente dürfen in einem Satz nur einmal programmiert sein. Eine Ausnahme bilden hierbei ”Schnelle Ausgänge” und ”Parameterzuweisung”. Diese dürfen mehrfach in einem Satz angegeben werden.

Rechenanweisungen werden mit einem Doppelpunkt eingeleitet, müssen alleine in einem Satz stehen und dürfen nur mit Kommentaren kombiniert werden. Ein Satz darf maximal 256 Zeichen lang sein.

Programmschleife und -verzweigung

Für Programmschleifen bzw. -verzweigungen können einer Anweisung Satznummern vorangestellt werden. Diese müssen innerhalb des Programms in aufsteigender Reihenfolge vergeben werden.

Programmende

Das Programmende wird durch eine M- oder G-Funktion gekennzeichnet:

- ► M30 = Hauptprogramm-Ende
- ► G99 = Unterprogramm-Ende

Kommentare

Kommentare werden durch runde oder geschweifte Klammern eingegrenzt und zur Laufzeit überlesen.

Ein Kommentar darf am Satzende oder allein in einem Satz stehen. Die Zeilenlänge von 256 Zeichen darf auch mit Kommentar nicht überschritten werden. Wird ein Kommentar mit einer Klammer geöffnet, wird das Zeilenende automatisch als Kommentarende angesehen.

Bei der Verwendung der geschweiften Klammern für Kommentare besteht die Möglichkeit, diese beim Einlesen über die Monitorschnittstelle in die Steuerung zu unterdrücken. Dies wird durch die Maschinenkonstante MK_NCPROG_OHNE_KOMMENTARE bestimmt. Die Kommentare werden dann nicht in der Steuerung gespeichert.

Bei der Verwendung von Rechenanweisungen, sind Kommentare generell in geschweifte Klammern zu setzen.

Beispiel:

G17 (Ebenenauswahl XY-Ebene)
{Dies ist auch ein Kommentar}

Wegbedingung G

Die Wegbedingung ”G” ist entweder nur in ihrem Satz gültig (nicht modale Funktion) oder ersetzt eine andere, vorher gültige Bedingung (modale Funktion).

Modale G-Funktionen sind in Gruppen zusammengefasst, innerhalb dieser ist zu einem Zeitpunkt nur eine Funktion gültig.

Modale Funktionen

Modale Funktionen bleiben solange gültig, bis sie durch eine neue modale Funktion der gleichen Gruppe abgelöst werden. Diese Funktionen sind in folgende Gruppen eingeteilt:

Gruppe	Funktion
a	Interpolationsart
b	Umschaltung zwischen Bahn- und Streckenbetrieb
c	Ebenenauswahl
d	Maßangaben
e	Werkzeugkorrektur
f	Koordinatenverschiebung
g	Spindeldrehzahl-Vereinbarung
h	Funktionen, die sich nicht gegenseitig aufheben, sondern durch sich selbst oder eine spezielle Ausschaltfunktion aufgehoben werden

Bei einem Unterprogrammaufruf werden die gültigen modalen Zustände der Funktionsgruppen a ... d gerettet. Im Unterprogramm ist es dann möglich, neue modale Funktionen zu aktivieren. Der Unterprogrammrücksprung setzt anschließend die modalen Funktionen auf den geretteten Zustand. Dieses gilt über alle acht möglichen Unterprogrammebenen.

Eine Ausnahme bildet ein Programmabbruch, der entweder durch Drücken der **<STOP>**-Taste oder durch einen Fehler ausgelöst wird. Hierbei wird die Steuerung in den Bahnsteuerungsbetrieb und ins Bezugsmaßsystem zurückgeschaltet.

Diese und weitere modale Funktionen können am Programmende oder nach einem Programmabbruch mit Hilfe der Maschinenkonstante MK_VOREINSTELLUNG in einen definierten Zustand gebracht werden .

Satzvorverarbeitung und Zeitsynchronisation

Bei der Abarbeitung eines CNC-Programms wird zwischen der Interpretation und der Ausführung des Programms unterschieden. Die Ausführung eines DIN-Satzes erfolgt im Allgemeinen nicht zeitgleich zu dessen Interpretation. Es ist im Gegenteil für viele Funktionen wichtig, dass die Sätze vorausschauend interpretiert werden. So ermittelt zum Beispiel die Funktion "Look-Ahead" (G60 X0) über die interpretierten Interpolationssätze ein optimales Geschwindigkeitsprofil, was nur möglich ist, wenn genügend viele Sätze im Voraus interpretiert werden.

Programmausführung und Interpretation erfolgen grundsätzlich parallel und sind nur über einen Vorlaufpuffer (FIFO-Speicher) gekoppelt, in den Interpolations- und andere Aufträge zur Ausführung eingetragen werden. Erst durch diese Entkopplung ist eine zügige Programmabarbeitung möglich. Die max. Anzahl der Sätze in diesem FIFO hat damit maßgeblichen Einfluss auf die Abarbeitungsgeschwindigkeit eines NC-Programms. Diese Anzahl kann mit der Maschinenkonstante MK_LAH_VORLAUFTIEFE auf bis zu 1024 Sätze eingestellt werden. Für die meisten Anwendungen reichen jedoch 16 Sätze vollkommen aus, für anspruchsvollere Anwendungen können auch 256 oder mehr Sätze sinnvoll sein.

An kritischen Programmstellen kann es notwendig sein, diese Satzvorverarbeitung zu stoppen, um die Interpretation mit der Ausführung der davor stehenden Sätze zu synchronisieren. Eine solche Zeitsynchronisation erfolgt automatisch, wenn im Programm auf einen zeitsynchronisierten Parameter im Parameterfeld zugegriffen wird oder eine zeitsynchronisierte G-Funktion programmiert ist.

Eine Zeitsynchronisation hat auch bei eingeschaltetem "Look-Ahead" immer einen kurzzeitigen Stillstand der Achsen auf der Bahn zur Folge, da ohne einen Nachfolgesatz keine Anschlussgeschwindigkeit bestimmt werden kann.

In der Tabelle der G-Funktionen sind die zeitsynchronisierten G-Funktionen durch den Index "s" markiert (📖 90).

Neben diesen Funktionen gibt es noch Befehle, die keine ausführende Funktion haben, sondern lediglich den sequenziellen Programmfluss beeinflussen oder auf andere Art nur die Interpretation des Programms beeinflussen. Diese Funktionen werden bereits bei der Interpretation vollständig ausgeführt, landen nicht im Vorlaufpuffer und führen daher nicht zu einer Unterbrechung der Bahnbewegung.

Zu diesen Befehlen gehören:

- Parameterfeldzuweisungen (P2000=100)
- Formeln (P2001=P2000+50)
- einige G-Funktionen (z. B. G20, G22)

Alle betroffenen G-Funktionen sind in der Tabelle der G-Funktionen durch einen Index markiert (📖 90).

3.2 G-Funktionen

Syntax

Ein DIN-Satz mit einer G-Funktion hat grundsätzlich folgenden Aufbau: Dem Buchstaben ”G” folgt die Nummer der G-Funktion. Daran anschließend folgen die Parameter, die jeweils durch ihren Adressbuchstaben und dem dazugehörigen Wert gebildet werden.

G(Nummer) [Adresskennzeichen(Adresswert)] ...

Mögliche Adresskennzeichen sind

- **X ,Y ,Z ,A ,B, C, U, V, W, x, y, z, a, b, c, u, v, w** als Kennzeichnungen für Achsen, im Folgenden als AXES bezeichnet.
- **D, F, H, I, J, K, L, R** als G-funktionsspezifische Fließpunktparameter
- **E, O** als G-funktionsspezifische Integerparameter

Als Adresswerte sind bei Achsadressen und Fließpunktparametern beliebige Fließpunktzahlen alternativ mit Exponent (z. B. -23.475e^{-3}) gültig. Bei allen anderen sind ganze Zahlen im Bereich von -32767 bis +32767 erlaubt.

Alternativ zu der direkten Eingabe können alle oben genannten Adressen auch indirekt programmiert werden. Dies wird durch die Angabe von ”P” angezeigt. Die Programmierung erfolgt über das P-Feld. Statt des Zahlenwertes wird dann der Index auf das P-Feld dem Adressbuchstaben zugeordnet.

Beispiel

G0 X=P1000	Für X gilt dann der Wert, der beim Index 1000 im P-Feld steht.
G0 X=PP1100	Für X gilt der Wert, dessen Index beim Index 1100 im P-Feld steht (doppelt indiziert).

Ein DIN-Satz muss nicht zwingend eine G-Funktion enthalten. Einige der Funktionen sind modal wirksam und müssen nicht wiederholt werden. Für den Rest des Satzes gelten dann die obigen Festlegungen.

3.2.1 G-Funktionen im Überblick

i Funktion ist eine reine Interpreterfunktion (📖 89)

s Funktion ist zeitsynchronisiert (📖 89)

1) Funktion wird bei Programmende oder -abbruch und bei modalen Verzweigungen deaktiviert

2) Funktion wird bei Programmende oder -abbruch deaktiviert

3) Funktion wird bei Programmabbruch deaktiviert

4) Funktion wird bei Programmabbruch und bei modalen Verzweigungen deaktiviert

Nr.	Bedeutung	Gruppe	Attribut
00	Geradeninterpolation im Eilgang	a	
01	Geradeninterpolation	a	
02	Kreisinterpolation im Uhrzeigersinn	a	
03	Kreisinterpolation im Gegenuhrzeigersinn	a	
04	Verweilzeit		
05	Spline-Interpolation über die programmierten Stützpunkte	h1)	i
06	Polynom-Interpolation mit Koeffizientenprogrammierung	h1)	i
10	Definition einer Wiederanlaufposition nach einem Fehler		

Nr.	Bedeutung	Gruppe	Attribut
16	Festlegen der Hauptachsen des aktuellen NC-Kanals	h	s
17	Ebenenauswahl (XY)	c	i
18	Ebenenauswahl (XZ)	c	i
19	Ebenenauswahl (YX)	c	i
20	Satzsprung, wahlweise mit Bedingungsprüfung		I
21	Synchrone Parameterfeldzuweisung		
22	Unterprogrammaufruf, wahlweise mit Bedingungsprüfung und Start eines neuen NC-Kanals		I
24	Negative Verfahrbereichsgrenze definieren	h	
25	Positive Verfahrbereichsgrenze definieren	h	
26	Verfahrbereichsgrenzen aktivieren/deaktivieren	h	
27	Satzsprung mit Wiederholungszähler		i
30	Streckenbetrieb	b	s
31	Bahnbetrieb	b	s
33	Kopplung zwischen Bahn und Rotationsachsen einschalten	h[2)]	
34	Bahnkopplung ausschalten	h	
36	Modales Oszillieren einschalten	h[3)]	s
37	Modales Oszillieren ausschalten	h	s
40	Ausschalten des Korrekturmoduls	e	i
41	Einschalten des Korrekturmoduls / links	e[4)]	i
42	Einschalten des Korrekturmoduls / rechts	e[4)]	i
53	Temporäre Koordinatenverschiebung aufheben	f	i
54	Temporäre Nullpunktverschiebung des aktuellen Werkstück-koordinatensystems	f[2)]	i
60	Genauhalt ein/aus	h[2)]	i
61	Satzvorverarbeitung anhalten		s
74	Referenzpunktfahrt		s
75	Maßstabsfaktor für Eingabeeinheiten ändern	h[2)]	i
76	Maßstabsfaktor für Impulsbewertung ändern	h[2)]	
88	Grunddrehung	h	s
89	Konturdrehung	h[2)]	s
90	Absolute Maßangaben (Bezugsmaß)	d	i
91	Inkrementale Maßangaben (Kettenmaß)	d	i
92	Relative Nullpunktverschiebung des akt. Koordinatensystems	f	i
93	Absolute Nullpunktverschiebung des akt. Koordinatensystems		
96	Programmierung der Spindelumfangsgeschwindigkeit	g	
97	Programmierung der Spindeldrehzahl		
99	Unterprogrammrücksprung		i
100	Polarkoordinaten: Geradeninterpolation im Eilgang	a	
101	Polarkoordinaten: Geradeninterpolation	a	
102	Polarkoordinaten: Kreisinterpolation im Uhrzeigersinn	a	
103	Polarkoordinaten: Kreisinterpolation gegen Uhrzeigersinn	a	
110	Polarkoordinaten: Mittelpunkt übernehmen		
110	Polarkoordinaten: Mittelpunkt übernehmen		i
112	Tangentiale Nachführung einschalten	h	i
113	Tangentiale Nachführung ausschalten	h	i
114	6-Achsen-Transformation (X,Y,Z,A,B,C)	h[2)]	s
115	Mantel-Transformation	h[2)]	s

Nr.	Bedeutung	Gruppe	Attribut
116	Rotationsachsen-Transformation (B,C) -> (A,B')	h[2)]	s
117	Reserviert		
118	Reserviert		
120	Koordinatenverschiebung über modalen Offset		s
121	Programmierung des modalen Offsets		s
122	Konfigurieren der Wirkung der Verfahrtasten	h	
125	Nichtmodale Vergleichsoperation		i
130	Modale Vergleichsoperation	h[2)]	s
131	Modale Vergleichsoperation austragen	h	s
132	Modale Programmverzweigung sperren	h[2)]	s
133	Modale Programmverzweigung erlauben	h	s
134	Nichtmodale Wartefunktion, Parameterfeldvergleich		i
140	Unrundschleifen ausschalten	E	S
141	Unrundschleifen mit neg. Drehrichtung einschalten	e[3)]	s
142	Unrundschleifen mit pos. Drehrichtung einschalten	e[3)]	s
143	Parametereinstellung der Unrundschleifphasen		i
144	Konfiguration einer Korrekturtabelle für Unrundschleifen	h	s
145	Festlegen der Korrekturwerte in der Korrekturtabelle	h	i
150	Modale Programmverzweigung auf ein externes Ereignis	h[2)]	s
151	Programmverzweigung auf ein externes Ereignis		s
152	Nichtmodale Wartefunktion, Q-Bit Vergleich		i
153	Warten auf die Terminierung eines NC-Kanals		
158	Intermittierenden Betrieb einschalten	h[4)]	
159	Intermittierenden Betrieb ausschalten	h	
161	Istpositionsübernahme		s
162	Achsverbund definieren		
175	Achstausch	h	s
180	Modales Fahren einschalten	h[3)]	s
181	Modales Fahren ausschalten	h	s
187	Handradkopplung ein-/ausschalten	h[3)]	
193	Absolutes Nullpunktsetzen im aktuellen Koordinatensystem	f	i
194	Grundoffsetverschiebung	f	s
195	Verschiebung aller Werkstückkoordinatensysteme außer S0	f	
200	Ein-/Ausschalten des Geometriefilters	h	i
201	Ändern der Beschleunigungs- und Bremsrampen	h[2)]	i
209	Setzen des Geometriezählers		i
211	Transformation für zweiachsige Knickarmroboter-Kinematik	h	s
222	Kundenspezifischen Ausgangssignalhandler aktivieren	h	
226	Wirkung der Hardwareendschalter umkonfigurieren	h	
231	Abstandsregelung im Achsrechner	h	s
232	Reserviert für Online-Messerkorrektur	h	s
233	3D-Achsenkorrektur	h	s
234	Richtungsabhängige Höhenkorrektur	h	s
250	Aufzeichnung von Achspositionen oder Geschwindigkeiten	h	
251	Sprungantwort einer Achse aufnehmen		s
252	Werteingabe über das Anzeigegerät		s
253	Textanzeigefunktion, optional mit Fehlerbehandlung		s

3.2.2 G-Funktionen-Einzelbeschreibungen

Im Folgenden wird jede G-Funktion detailliert beschrieben. Die Funktionen sind in numerisch aufsteigender Folge sortiert.

Hinweis!

Bei den Wegbedingungen sind explizit die Kennbuchstaben für die einzelnen Achsen angegeben. Diese beziehen sich auf eine Konfiguration mit maximal vier Achsen. Bei einer anderen Konfiguration können auch andere Achsbuchstaben verwendet werden. Diese sind dann sinngemäß in den Wegbedingungen zu verwenden. Die verwendeten Achsen mit den zugehörigen Kennbuchstaben werden in den Maschinenkonstanten festgelegt.

3.2.2.1 G00 Lineare Zustellung im Eilgang

Mit der Wegbedingung G00 wird eine lineare Zustellung mit Eilgang-Geschwindigkeit programmiert.

Abweichend von der DIN66025 kann im Streckenbetrieb oder im Bahnbetrieb verfahren werden. Dies wird durch G30 / G31 vorgewählt.

Syntax

G0 AXES R D E L

Bedeutung der Adressen

AXES	Zielpunktkoordinaten der Achsen
R	Radius, mit dem die nachfolgende Linearinterpolation (G0, G1) verbunden werden soll.
D	Max. Bahnabweichung im Zielpunkt zum Eckenverschleifen mit der nachfolgenden Linearinterpolation.
F	Eilganggeschwindigkeit auf der Bahn (G31) bzw. jeder Achse (G30).
E, L	Auswahl der Vorschubgeschwindigkeit über Drehzahl (E) und Schrittweite (L). F=E*L.

Erläuterung

G00 kann im Bahnbetrieb (G31) und im Streckenbetrieb (G30) programmiert werden.

Bahnbetrieb: Alle Achsen erreichen den programmierten Zielpunkt gleichzeitig. Zwei direkt aufeinanderfolgende G0/G1-Funktionen können mit einem tangentialen Kreisbogen mit dem Radius R verbunden werden. Dazu ist im ersten Satz der Parameter ”R” anzugeben. Der programmierte Radius wird seitens der Steuerung hin zu kleineren Radien korrigiert, wenn der Abstand zwischen dem Gerade-Kreis-Schnittpunkt und dem Scheitelpunkt die Länge einer der beiden Geraden überschreitet oder im nachfolgenden Satz ebenfalls ein R programmiert ist. Im letztern Fall wird die zur Verfügung stehende Bahnlänge für die Kreiseinpassung auf die Hälfte gekürzt.

Alternativ zu R kann mit D die maximale Bahnabweichung im Zielpunkt programmiert werden, um die zwischen zwei G0/G1-Sätzen entstehende Ecke definiert zu verschleifen. Auch hier müssen die beiden Sätze direkt hintereinander programmiert sein, ansonsten wird das D ignoriert. Im Gegensatz zum Eckenrunden mit R muss beim Eckenverschleifen keine Ebene ausgewählt werden, da zwischen beiden Sätzen kein Kreis, sondern ein Polynom (G6) eingefügt wird. Die Interpolation erfolgt im Raum.

Die bei F programmierte Eilgangsgeschwindigkeit ist modal für G0 wirksam, sie hat keinen Einfluss auf die Geschwindigkeit bei G1, G2 oder G3. Voreingestellt ist der Wert aus MK_VBAHNMAX.

Die bei E und L programmierten Werte sind modal wirksam und haben keinen Einfluss auf die Eilgangsgeschwindigkeit, sondern nur auf G1, G2 und G3. Die resultierende Vorschubgeschwindigkeit wird jedoch nur berücksichtigt, wenn bei L ein Wert ungleich 0 programmiert wurde. Der bei E programmierte Wert wird zusätzlich zeitsynchron in P561 abgelegt und kann z. B. als Spindeldrehzahl verwendet werden.

Die Geschwindigkeit wird so begrenzt, dass weder die max. Bahngeschwindigkeit (MK_VBAHNMAX) noch die max. Achsgeschwindigkeiten (MK_VMAX) überschritten werden. Die Beschleunigungs- und Bremsrampen auf der Bahn werden so begrenzt, dass die max. Rampen (MK_BESCHL, MK_BREMS) der beteiligten Achsen nicht überschritten werden.

Streckenbetrieb: Alle Achsen erreichen ihren Zielpunkt unabhängig voneinander. Der nachfolgende Satz wird erst dann ausgeführt, wenn alle Achsen ihren Zielpunkt erreicht haben.

Die bei F programmierte Eilgangsgeschwindigkeit ist modal für G0 und alle programmierten Achsen wirksam, sie hat keinen Einfluss auf die Geschwindigkeit bei G1, G2 oder G3. Voreingestellt sind die Werte aus MK_VMAX.

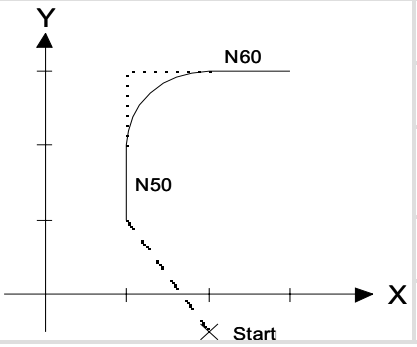
Die Geschwindigkeiten der beteiligten Achsen werden auf die jeweiligen max. Achsgeschwindigkeiten (MK_VMAX) in den Maschinenkonstanten begrenzt. Als Beschleunigungs- und Bremsrampen werden die in den Maschinenkonstanten konfigurierten Maximalwerte (MK_BESCHL, MK_BREMS) verwendet, falls nicht mit G201 kleinere Werte gewählt wurden.

Generell: Die Wegbedingung G00 ist modal wirksam.

Unter den Adressen der Positionierachsen kann die Endposition der Achsen entweder im Bezugsmaßsystem (G90) oder im Kettenmaßsystem (G91) programmiert werden.

Die Einheit von F ist davon abhängig, welche Achsen programmiert sind. Falls nur Linearachsen beteiligt sind, wird in F in Eingabeeinheiten/min (Default: mm/min) programmiert. Sind nur Rotationsachsen beteiligt, wird F in 1/min programmiert. Wenn beide beteiligt sind, ist die Einheit von F davon abhängig ob die Führungsachsen Linear oder Rotationsachsen sind (siehe G16).

Beispiel



N10 G90	Ins Bezugsmaßsystem umschalten und die X-Achse auf die Position +20 mm und die Z-Achse die Position +200 mm verfahren.
N20 G0 X20 Z200	
N50 G0 X10 Y10	Modale G-Funktion auswählen und Startposition anfahren.
N60 Y30 R10	Die Sätze N60 und N70 werden statt mit einem 90° Winkel mit einem tangentialen Kreisbogen vom Radius 10 mm verbunden.
N70 X30	

3.2.2.2 G01 Geradeninterpolation

Bei der linearen Interpolation bewegt sich das Werkzeug zwischen dem Startpunkt, d. h. der aktuellen Ist-Position und dem programmierten Endpunkt (Sollposition) auf einer Geraden. Der programmierte Endpunkt wird von allen Achsen gleichzeitig erreicht.

Syntax

G1 AXES R D F E L

Bedeutung der Adressen

AXES	Zielpunktkoordinaten der Achsen.
R	Radius, mit dem die nachfolgende Linearinterpolation (G0, G1) verbunden werden soll.
D	Max. Bahnabweichung im Zielpunkt zum Eckenverschleifen mit der nachfolgenden Linearinterpolation.
F	Vorschubgeschwindigkeit auf der Bahn (G31) bzw. jeder Achse (G30).
E,L	Auswahl der Vorschubgeschwindigkeit über Drehzahl (E) und Schrittweite (L). F=E*L.

Erläuterung

G01 kann im Bahnbetrieb (G31) und im Streckenbetrieb (G30) programmiert werden.

Bahnbetrieb: Alle Achsen erreichen den programmierten Zielpunkt gleichzeitig.

Zwei direkt aufeinanderfolgende G0/G1-Funktionen können durch einen tangentialen Kreisbogen mit dem Radius R verbunden werden. Dazu ist im ersten Satz der Parameter R anzugeben. Der programmierte Radius wird seitens der Steuerung hin zu kleineren Radien korrigiert, wenn der Abstand zwischen dem Gerade-Kreis-Schnittpunkt und dem Scheitelpunkt die Länge einer der beiden Geraden überschreitet oder im nachfolgenden Satz ebenfalls ein R programmiert ist. Im letzten Fall wird die zur Verfügung stehende Bahnlänge für die Kreiseinpassung auf die Hälfte gekürzt.

Alternativ zu R kann mit D die maximale Bahnabweichung im Zielpunkt programmiert werden, um die zwischen zwei G0/G1-Sätzen entstehende Ecke definiert zu verschleifen. Auch hier müssen die beiden Sätze direkt hintereinander programmiert sein, ansonsten wird das D ignoriert. Im Gegensatz zum Eckenrunden mit R muss beim Eckenverschleifen keine Ebene ausgewählt werden, da zwischen beiden Sätzen kein Kreis, sondern ein Polynom (G6) eingefügt wird. Die Interpolation erfolgt im Raum.

Die bei F programmierte Vorschubgeschwindigkeit ist modal wirksam. Voreingestellt ist der Wert 0.

Die bei E und L programmierten Werte sind modal wirksam. Die resultierende Vorschubgeschwindigkeit wird jedoch nur berücksichtigt, wenn bei L ein Wert ungleich 0 programmiert wurde. Der bei E programmierte Wert wird zusätzlich zeitsynchron in P561 abgelegt und kann z. B. als Spindeldrehzahl verwendet werden.

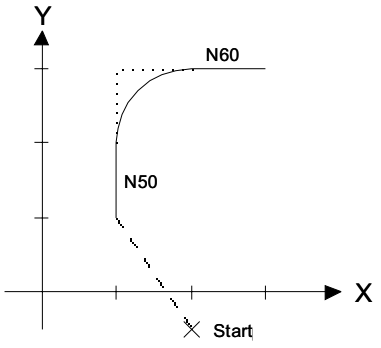
Die Geschwindigkeit wird so begrenzt, dass weder die max. Bahngeschwindigkeit (MK_VBAHNMAX) noch die max. Achsgeschwindigkeiten (MK_VMAX) überschritten werden. Die Beschleunigungs- und Bremsrampen auf der Bahn werden so begrenzt, dass die max. Rampen (MK_BESCHL, MK_BREMS) der beteiligten Achsen nicht überschritten werden.

Streckenbetrieb: Alle Achsen fahren mit der programmierten Geschwindigkeit und erreichen ihren Zielpunkt unabhängig von einander entsprechend der Verfahrstrecke und der Geschwindigkeit. Der nachfolgende Satz wird erst ausgeführt, wenn alle Achsen ihren Zielpunkt erreicht haben. Die bei F programmierte Vorschubgeschwindigkeit ist modal für alle programmierten Achsen wirksam. Voreingestellt ist für alle Achsen der Wert 0. Die Geschwindigkeiten der beteiligten Achsen werden auf die jeweiligen max. Achsgeschwindigkeiten (MK_VMAX) in den Maschinenkonstanten begrenzt. Als Beschleunigungs- und Bremsrampen werden die in den Maschinenkonstanten konfigurierten Maximalwerte (MK_BESCHL, MK_BREMS) verwendet, falls nicht mit G201 kleinere Werte gewählt wurden.

Generell: Die Wegbedingung G01 ist modal wirksam.

Unter den Adressen der Positionierachsen kann die Endposition der Achsen entweder im Bezugsmaßsystem (G90) oder im Kettenmaßsystem (G91) programmiert werden. Die Einheit von F ist davon abhängig welche Achsen programmiert sind. Falls nur Linearachsen beteiligt sind, wird in F in Eingabeeinheiten/min (Default: mm/min) programmiert. Sind nur Rotationsachsen beteiligt, wird F in 1/min programmiert. Wenn beide beteiligt sind, ist die Einheit von F davon abhängig ob die Führungsachsen Linear oder Rotationsachsen sind (siehe G16).

Beispiel



N10 G31	Bahnbetrieb einschalten
N20 G1 X0 Z10 C0 F2	X-Achse auf 0 mm und Z-Achse auf +10 mm mit 2 mm/min positionieren, gleichzeitig dazu wird die C-Achse auf 0° positioniert.
N30 G1 C360 F60	C-Achse bewegt sich auf die Position 360 Grad mit der Geschwindigkeit 60 Umdrehungen pro Minute.
N50 G1 X10 Y10	Modale G-Funktion auswählen und Startposition anfahren.
N60 Y30 R10 N70 X30	Die Sätze N60 und N70 werden statt mit einem 90° Winkel mit einem tangentialen Kreisbogen vom Radius 10 mm verbunden.

3.2.2.3 G02 Kreisinterpolation im Uhrzeigersinn, G03 Kreisinterpolation gegen Uhrzeigersinn

Definition eines Kreises bzw. Kreisabschnittes im Uhrzeiger- bzw. Gegen-Uhrzeigersinn, mit linearer Zustellung einer Linearachse (Schraubenlinie) oder aufweitendem Radius (Spirale) mit zusätzlicher Zustellung einer Linearachse (konische Schraubenlinie)

Hinweis!

Bei der Verwendung von Korrekturmodulen (z. B. WRK) kann nur der normale Kreisbogen programmiert werden. Die Zusatzfunktionen sind nicht erlaubt!

Syntax

G02 AXES D I J K R F E L

G03 AXES D I J K R F E L

Bedeutung der Adressen

AXES	Zielpunktkoordinaten der drei linearen Hauptachsen (Default X,Y,Z), sowie Start- bzw. Zielpunktkoordinaten der drei rotativen Nebenachsen (Default A,B,C).
D	Radiusänderung bei Erreichen des Zielpunktes
I	Mittelpunktkoordinate der ersten Hauptachse (X) oder Anzahl zusätzlicher Vollkreise
J	Mittelpunktkoordinate der zweiten Hauptachse (Y) oder Anzahl zusätzlicher Vollkreise
K	Mittelpunktkoordinate der dritten Hauptachse (Z) oder Anzahl zusätzlicher Vollkreise
R	Interpolationsradius
F	Bahngeschwindigkeit
E, L	Auswahl der Vorschubgeschwindigkeit über Drehzahl (E) und Schrittweite (L). F=E*L.

Erläuterung

Mit G02/G03 wird die Interpolation auf einer Kreisbahn in der ausgewählten Ebene (G17, G18 bzw. G19) programmiert (G02 im Uhrzeigersinn, G03 gegen den Uhrzeigersinn).

Hinweis!

In jedem Fall muss eine Ebene ausgewählt sein, es gibt keine standardmäßige Voreinstellung!

Für die Bestimmung eines Kreisbogens ist die Angabe von 3 Punkten notwendig. Dies sind in jedem Fall der Kreisbogenstartpunkt und Kreisbogenendpunkt, sowie entweder ein Mittelpunkt oder ein Radius. Somit ergeben sich für die Bestimmung eines Kreises, je nach Ebene, folgende zwingende DIN-Adressen Kombinationen

G17: ”XY IJ” oder ”XY R”
G18: ”ZX KI” oder ”ZX R”
G19: ”YZ JK” oder ”YZ R”

Die Angabe der Zielwerte kann sowohl im Absolutmaß als auch im Kettenmaß erfolgen.

Für fehlende Zielwerte werden die entsprechenden Kreisstartwerte eingesetzt. Die Angabe eines Mittelpunktes wird immer als Relativangabe zum Kreisstartpunkt interpretiert. Da sich bei der Radiusprogrammierung rechnerisch immer mehr als ein Kreismittelpunkt ergibt, ist in diesem Modus der Kreisbestimmung die Programmierung eines Vollkreises nicht möglich. Ein negativer Radius wählt von den beiden möglichen Mittelpunkten denjenigen aus, der den größeren Kreisbogen zur Folge hat. Durch Kombination der über die notwendigen Angaben hinausgehenden DIN-Adressen der G2/G3, ergeben sich folgende Geometrien:

a) Schraubenlinie mit Zylinderoberfläche

Programmiert werden muss die senkrecht zur ausgewählten Ebene stehende Linearachse. Diese Angabe bewirkt eine gleichzeitige Linearinterpolation der Achse zwischen Kreisstartpunkt und Kreiszielpunkt. Ist keine zusätzliche optionale DIN-Adresse angegeben, dann überstreicht die Schraubenlinie maximal 360°.

G17: Z
G18: Y
G19: X

Durch Angabe der Anzahl (n) zusätzlicher Vollkreise lässt sich die Schraubenlinie auf maximal (n+1)*360° ausdehnen.

G17: ZK
G18: YI
G19: XJ

b) Schraubenlinie mit Konusoberfläche

Hierbei gelten alle Angaben wie unter a). Zusätzlich muss aber ein Delta-Radius (**D**) programmiert werden. Der Konusradius öffnet (D>0) oder schließt (D<0) sich dann gleichmäßig vom Kreisstartpunkt bis zum Kreiszielpunkt.

G17: ZKD
G18: YID
G19: XJD

c) Spirale in der Ebene

Alle Angaben wie unter b) allerdings ohne die Linearzustellung werden benötigt.

G17: KD
G18: ID
G19: JD

Durch vorheriges Ausführen der Wegbedingung G112 ist die tangentiale Nachführung der Rotationsachse möglich, die der angewählten Ebene (Default: G17: C, G18: B, G19: A) zugeordnet ist. Wurde die G112 mit "Schlagzustellung der Rotationsachse" programmiert, ist bei der G2/G3 die Angabe des Startwinkels der Rotationsachse unter den entsprechenden DIN-Adressen (A,B,C) möglich. Wenn die tangentiale Nachführung nicht eingeschaltet ist, darf der Zielwinkel der Rotationsachse programmiert werden. Die Rotationsachse wird dann linear zwischen Start- und Zielwinkel auf der Kreisbahn zugestellt.

Die dem aktuellen NC-Kanal zugeordneten Haupt- und Nebenachsen können mit G16 ausgewählt werden.

Die Bahngeschwindigkeit kann unter der Adresse F programmiert werden. Wenn die Geschwindigkeit nicht programmiert wird, ist die zuletzt programmierte gültig.

Die Bahngeschwindigkeit wird in Abhängigkeit vom Radius und der erlaubten Bahnbeschleunigung laut folgender Formel begrenzt:

$F_{max} = \sqrt{3,6 * R * B} * 1000$ [mm/min] mit R=Radius [mm] und B=Beschleunigung [m/s^2]

Die bei E und L programmierten Werte sind modal wirksam. Die resultierende Vorschubgeschwindigkeit wird jedoch nur berücksichtigt, wenn bei L ein Wert ungleich 0 programmiert wurde. Der bei E programmierte Wert wird zusätzlich zeitsynchron in P561 abgelegt und kann z. B. als Spindeldrehzahl verwendet werden.

Besonderheiten bei der Kreisberechnung:

Bei Verwendung eines veränderlichen Radius (D), interpretiert die Steuerung die programmierte Geschwindigkeit als Vorgabe für den größeren Radius. Die Winkelgeschwindigkeit ist über die komplette G2/G3 Kontur konstant, während die programmierte Bahngeschwindigkeit sich auf dem Kreisbogen nur bei Erreichen des größeren Radius einstellt. Ist eine tangentiale Nachführung programmiert, wird die Bahngeschwindigkeit dem Beschleunigungsvermögen der nachgeführten Achse angepasst. Die Geschwindigkeits- und Beschleunigungswerte der optionalen Linearachse werden nicht berücksichtigt.

Die bei der Berechnung der Kreisgeometrien eventuell auftretenden Ungenauigkeiten, die in dem über Maschinenkonstanten definierten Fenster liegen, werden von der Steuerung über den Deltaradius kompensiert. Mit anderen Worten, alle Koordinatenangaben bei G2/G3 (Ziel- und Mittelpunkte) werden als nicht zu korrigierende Vorgabe interpretiert.

Beispiel

Alle nachfolgenden Beispiele zur G2/G3-Programmierung setzen voraus, dass zuvor die XY-Ebene (G17) eingeschaltet wurde.

N10 G0 X50 Y100	Kreisstartpunkt anfahren
N20 G2 Y200 I0 J50 F200	Fährt einen Halbkreis im Uhrzeigersinn von X50 Y100 nach X50 Y200 mit einem Kreismittelpunkt von X50 Y150, einem Radius von 50 mm und einer Bahngeschwindigkeit von 200 mm/min.
N30 G3 X-50 Y100 R100	Fährt einen Viertelkreis gegen den Uhrzeigersinn von X50 Y200 nach X-50 Y100 mit einem Radius von 100 mm.
G0 X0 Y0	Startposition X0 Y0
G3 X40 Y40 R100 D-60 K2	Fährt eine Spirale gegen den Uhrzeigersinn mit einem Startradius von 100 mm, einem Endradius von 40 mm (R+D) und einem überstrichenen Winkel von 810°.
G0 X0 Y0 Z0	Startposition X0 Y0 Z0
G3 X10 Y0 Z115 K12 R10	Fährt eine Schraubenlinie auf einer Zylinderfläche gegen den Uhrzeigersinn mit einem Radius von 10 mm, einer Länge von 115 mm (Y) und einem überstrichenen Winkel von 4500°.
G0 X0 Y-30 Z-115	Startposition X0 Y-300 Z-115
G3 X30 Y0 Z0 K10 R10 D20	Fährt eine Schraubenlinie auf einer Konusoberfläche gegen den Uhrzeigersinn mit einem Startradius von 10 mm, einem Endradius von 30 mm (R+D), einer Länge von 115 mm (Y) und einem überstrichenen Winkel von 3690°.

3.2.2.4 G04 Verweilzeit

Mit der Wegbedingung G04 wird die Verweilzeit programmiert, d. h., dass die Maschine sich für den programmierten Zeitraum nicht bewegt.

Syntax

G04 X

Bedeutung der Adressen

X: Verweilzeit in Sekunden

Erläuterung

Die Verweilzeit wird unter dem Adressbuchstaben ”X” programmiert, und zwar in Schritten von 0,01 s; die Einheit ist 1 Sekunde.

Die minimale Zeit ist vom Grobinterpolationstakt abhängig, die maximale Zeit beträgt 100 Minuten (6000 s).

Die vor der Verweilzeit gültigen modalen Wegbedingungen bleiben wirksam. In einem Satz mit der Wegbedingung G04 darf noch eine M-Funktion programmiert werden, die nach der Verweilzeit bearbeitet wird.

Beispiel

N30 G4 X5.1

Die Verweilzeit von 5,1 s warten, danach mit dem nachfolgenden Satz fortfahren.

3.2.2.5 G05 Spline - Interpolation

Die Spline-Interpolation wird mit G05 und der Angabe der beteiligten Achsen und ggf. weiterer Parameter eingeschaltet.

Syntax

G05 AXES I J K L

Bedeutung der Parameter

AXES	Gültigkeitskennung (L=0) oder Startposition des letzten Satzes (L=1)
I	Randbedingungen (Kopplung an benachbarte Raumkurven) 0: Freie Ränder (Default) 1: Anfang stetig und Ende mit freiem Rand 2: Anfang freier Rand, Ende stetig 3: Stetige Ränder
K	Wahl des Polynomgrades
	0: Polynom 3. Grades (zweite Ableitung nicht stetig, Default) 1: Polynom 5. Grades (zweite Ableitung stetig)
J	Tangentiale Nachführung der Rotationsachse 0: Relativ zu der Startposition der Achse (Default) 1: Zustellung der Rotationsachse im Startpunkt der Spline-Kurve auf den Tangentenwinkel
L	Kennung 0: Offene Kurve (Default) 1: Geschlossene Kurve

Erläuterung

Die Stützpunkte werden durch die nachfolgende Angabe der Koordinaten der Achsen angegeben. Die Anzahl der Stützpunkte ist beliebig. Das Ende eines Konturzugs, der mit der Spline-Interpolation durchfahren werden soll, wird durch eine G05 ohne Parameter gekennzeichnet. Nachfolgende Positionen werden dann in der programmierten Interpolationsart angefahren. Ein Spline, der sich nur über einen Satz erstreckt, wird als normaler Linearsatz ausgeführt.

Die an der Spline-Interpolation beteiligten Achsen werden durch Angabe einer Gültigkeitskennung bei dem jeweiligen Achsbuchstaben gekennzeichnet. Alle übrigen Achsen, die in den nachfolgenden Sätzen programmiert sind, werden linear interpoliert. Sie kommen zeitgleich mit den Spline-Achsen an ihrem Ziel an.

Wenn eine geschlossene Kurve im Start- und gleichzeitig Zielpunkt stetig sein soll, muss die Steuerung bereits zu Beginn der Kontur den Startpunkt des letzten Satzes kennen. Dieser muss dann explizit unter den Achsadressen bei der G05 angegeben werden und die Kennung L auf 1 gesetzt werden.

Innerhalb einer Spline-Folge werden die Stützstellenableitungen mit Hilfe der benachbarten Stützstellen berechnet. Am linken und rechten Rand einer Spline-Folge kann die Ankopplung an die benachbarten Konturabschnitte über die Adresse I beeinflusst werden. Es sind freie Ränder und die stetige Ankopplung zu unterscheiden, wobei eine stetige Verbindung mit zirkularen Abschnitten nur in einer Hauptebene erfolgen kann.

Bei einer stetigen Ankopplung am Ende muss der G05 ein Linear- oder Zirkularsatz folgen. Die Beeinflussung der Rotationsachse im Startpunkt der Spline-Kurve mit der Adresse J gilt nur dann, wenn vorher die tangentiale Nachführung der Achse mit G112 eingeschaltet war.

Hinweise für die Verwendung der Spline-Interpolation:

Die Spline-Interpolation ist auch im Zusammenhang mit einer Werkzeugradiuskorrektur (WRK) zulässig. Zu beachten ist dabei, dass die WRK vorher ausgeführt wird, d. h. die Spline-Interpolation findet über die korrigierten Stützpunkte statt.

Eine Spline-Interpolation über Stützpunkte, die im Polarkoordinatensystem programmiert sind, ist möglich.

Der verwendete Algorithmus für die Spline-Interpolation erlaubt auch die Rückwärtsinterpolation der Sätze.

In einem Satz mit Stützpunkten dürfen gleichzeitig eine M-Funktion und schnelle Ausgänge programmiert werden.

Die Verwendung modaler Vergleichsoperationen ist möglich. Diese dürfen jedoch nicht innerhalb der Spline-Sequenz programmiert werden. Bei einem Abbruch wird die Spline-Interpolation abgeschaltet. Die Verwendung eines Interruptprogramms ist möglich. Die programmierten Punkte werden in den jeweiligen Interpolationsarten angefahren. Die Verwendung der Spline-Interpolation im Interrupt-Programm ist nicht zulässig.

Bei der indirekten Programmierung der Stützpunkte ist zu beachten, dass nicht auf zeitsynchronisierte Parameter (z. B. aktuelle Ist-Position) zugegriffen werden kann.

Ein Wechsel zwischen Absolutmaß und Kettenmaß ist möglich. S-/T-Umschaltungen und Koordinatenverschiebungen sind erlaubt. Sprungfunktionen (G20, G22, G125) sind ebenfalls erlaubt.

Direkt aufeinanderfolgende Sätze mit identischen Stützpunkten werden ignoriert. Falls Zusatzfunktionen programmiert sind, gehen diese ebenfalls verloren.

3.2.2.6 G06 Polynom - Interpolation

Polynom-Interpolation dritten Grades in einem Satz.

Syntax

G06 AXES[Ziel;a_2;a_3] NAXS I J K E L F

Bedeutung der Parameter

AXES	Beliebige Achsadresse, deren Polynom-Koeffizienten (Ziel, a2, a3) programmiert werden sollen.
NAXS	Nebenachse, die linear zu der Bahnbewegung interpoliert wird (a2=0, a3=0)
I	Lage des Segments im Polynomzug 0: Zwischensegment (Default) 1: das erste Segment 2: das letzte Segment 3: nur ein Segment
J	Tangentiale Nachführung der Rotationsachse 0 oder nicht programmiert - relativ zu der Startposition der Achse 1: die Rotationsachse wird zunächst im Startpunkt dieses Segmentes auf den Tangentenwinkel zugestellt
K	Art der Kurve - nur mit erstem Segment der Kurve programmiert 0: eine offene Kurve (Default) 1: eine geschlossene Kurve
F	Bahnvorschub
E, L	Auswahl der Vorschubgeschwindigkeit über Drehzahl (E) und Schrittweite (L). F=E*L.

Erläuterung

Mit G6 wird für jede Achse ein Polynom dritter Ordnung der Form

$f(p) = a_3p^3 + a_2p^2 + a_1p + a_0$

programmiert. Dabei werden außer der Zielposition der jeweiligen Achse noch die Koeffizienten a_2 und a_3 angegeben. Der Koeffizient a_0 entspricht der Startposition der Achse und a_1 wird intern berechnet. p ist die Position auf der Sehne, die sich zwischen Start und Zielposition der beteiligten Achsen ergibt.

Beispiel

G06 X[100;0.1234;-0.00182] Y[15;0.00145;-0.0003067] A30 I3 F2000

3.2.2.7 G10 Definition einer Wiederanlaufposition

Mit Hilfe der G10 können im Programm Wiederanlaufpositionen gesetzt werden, an denen die Bearbeitung nach einem Fehler wieder aufgenommen werden kann.

Syntax

G10 AXES

Bedeutung der Adressen

AXES	Achsen, die für den Wiederanlauf repositioniert werden sollen.

Erläuterung

Nach einem leichten Fehler kann mit Hilfe des ”Start nach G10”-Befehls (siehe ”MC- und NC Software Handbuch”) die Bearbeitung des Programms, an der letzten programmierten G10 vor dem Fehler, wiederaufgenommen werden. Die G10 speichert dazu folgende Informationen:

- ► Absolut-/Relativmaß (G90/G91)
- ► Bahn-/Streckenbetrieb (G31/G30)
- ► Hauptebene (G17,G18,G19)
- ► Modale G-Funktion (G0,G1,G2,G3)
- ► Unterprogrammebene
- ► Werkzeug- und Werkstückkoordinatensystem (T,S)
- ► Ist-Positionen der programmierten Achsen
- ► Status des Korrekturmoduls (G40,G41,G42)

Bitte beachten Sie, dass die G10 weder den Zustand des Parameterfeldes noch die Nullpunkte der Koordinatensysteme speichert, auch der Zustand anderer NC-Funktionen wie Spline (G5) und Radiusübergang (G1 mit R) werden nicht gerettet.

Verwenden Sie die G10 nach Möglichkeit nur am Anfang abgeschlossener Konturabschnitte.

Für die Funktion ”Wiederanlauf nach Fehler” muss die Maschinenkonstante MK_FEHLERRESTART gesetzt werden, sonst erfolgt auch nach einem leichten Fehler ein Fehlerabbruch und die Bearbeitung kann nicht an der Unterbrechungsstelle wiederaufgenommen werden.

Beispiel

N80 G10 X0 Y0	Satz 80 wird als Wiederanlaufsatz markiert und die aktuellen Positionen der X- und Y-Achse als Wiederanlaufposition gespeichert.

3.2.2.8 G16 Auswahl der Haupt- und Nebenachsen des aktuellen NC-Kanals

Mit der G16 können die an den drei Hauptebenen beteiligten Haupt- und Nebenachsen des aktuellen NC-Kanals beliebig ausgewählt und damit die voreingestellten Achsen X,Y,Z und A,B,C ersetzt werden.

Syntax

G16 AXES

Bedeutung der Adressen

AXES	Achsen, die als Haupt- und Nebenachsen ausgewählt werden sollen. 1: erste Hauptachse (X) 2: zweite Hauptachse (Y) 3: dritte Hauptachse (Z) 4: erste Nebenachse (C) 5: zweite Nebenachse (A) 6: dritte Nebenachse (B)

Erläuterung

Die Auswahl anderer Haupt- und Nebenachsen wird notwendig, wenn andere als die voreingestellten Achsen mit G2/G3 interpoliert, mit Hilfe der WRK korrigiert oder beim Look-Ahead berücksichtigt werden sollen.

Wenn die erste Hauptachse eine Linearachse ist, dann muss die zweite und dritte Hauptachse ebenfalls eine Linearachse und alle Nebenachsen Rotationsachsen sein. Alle Achsen, die an den drei Hauptebenen beteiligt werden sollen, müssen programmiert werden. Die Programmierung der Vorschubgeschwindigkeit erfolgt bei allen Interpolationsaufgaben, an denen eine der Hauptachsen oder eine andere Linear-Achse beteiligt ist, in mm/min.

Ist die erste Hauptachse eine Rotationsachse, dürfen keine weiteren Achsen bei G16 programmiert werden. Die programmierte Achse wird zur Führungsachse. Die Programmierung der Vorschubgeschwindigkeit bei allen Interpolationen, an denen diese Achse beteiligt ist, erfolgt in 1/min.

Die G16 kann ohne Parameter programmiert werden, um die Default-Konfiguration mit X,Y,Z und A,B,C wieder herzustellen.

Beispiel

G16 U1 V2 W4	Auswahl der U- und V-Achse als lineare Hauptachsen und W als rotative Nebenachse, gültig in der X-Y-Ebene (G17).
G17	U-V-Ebene auswählen.
G0 U0 V0	Startpunkt für Kreisinterpolation anfahren.
G2 U100 R50	Halbkreis in der U-V-Ebene interpolieren.
G16	Default-Konfiguration wiederherstellen.

3.2.2.9 G17, G18, G19 Ebenenauswahl

Über die Wegbedingungen G17 ... G19 wird für diverse Funktionen, wie z. B. Kreisinterpolation in 2 Achsen die entsprechende Ebene angewählt.

Syntax

G17
G18
G19

Erläuterung

Ausgewählte Ebene:

G17	X-Y-Ebene
G18	X-Z-Ebene
G19	Y-Z-Ebene

Die ausgewählte Ebene ist im Programm modal gültig. Nach Programmende ist automatisch ”G17” gültig.

Beim ersten Start eines Programms muss unbedingt eine Ebene angewählt werden. Eine standardmäßige Voreinstellung ist nicht vorgesehen.

Beispiel

G18	Die Ebene Z-X wird angewählt.

3.2.2.10 G20 Satzsprung, wahlweise mit Bedingungsprüfung

Programmverzweigung innerhalb eines Programms mit Programmierung des Sprungziels und einer optionalen Sprungbedingung.

Syntax

G20 X E

Bedeutung der Adressen

X	Sprungziel
E	Index des Q-Bits (direkte Programmierung) bzw. des Parameterfeldes (indirekte Programmierung) zur Bedingungsprüfung.

Erläuterung

Mit der G20 wird ein Sprung innerhalb eines Programms programmiert. Ist zusätzlich die DIN-Adresse E angegeben, so wird die Ausführung des Sprunges von einer Bedingung abhängig gemacht.

Ist E direkt programmiert, wird die Funktion zeitsynchronisiert und der Wert von E als Index eines Q-Bits interpretiert. Der Sprung wird nur ausgeführt, wenn das Q-Bit ”1” ist.

Im Falle indirekter Programmierung wird der Sprung nur ausgeführt, wenn der Inhalt des angegebenen Parameters ungleich ”0” ist. Eine Zeitsynchronisation erfolgt nur, wenn es sich um einen zeitsynchronisierten Parameter handelt.

Beispiel

N20 G20 X50	Nach dem Satz 20 wird der Satz 50 bearbeitet.
G20 X1 E14	Fahre fort mit Satz 1, wenn das Q-Bit 14 einen Wert ungleich 0 enthält. Sonst fahre mit dem Satz nach der G20 fort.

3.2.2.11 G22 Unterprogrammaufruf, wahlweise mit Bedingungsprüfung und Start eines neuen NC-Kanals

Aufruf eines Unterprogramms als separates Programm oder innerhalb des aktuellen Programms, wahlweise von einer Bedingung abhängig und mit Wiederholung programmierbar. Das Unterprogramm kann auch in einem neuen NC-Kanal gestartet werden.

Syntax

G22 AXES K L J I E

Bedeutung der Adressen

AXES	Achsen, die an den neuen NC-Kanal (K) übergeben werden sollen.
K	Nummer des neuen NC-Kanals (> 1)
L	Nummer des Unterprogramms {1 ... 9999}
J	Satznummer
I	Schleifenzähler
E	Index des Q-Bits (direkte Programmierung) bzw. des Parameterfeldes (indirekte Programmierung) zur Bedingungsprüfung.

Erläuterung

Die Funktion G22 erlaubt einen Unterprogramm-Aufruf, der auf Wunsch auch über die DIN-Adresse E bedingt ausführbar ist und über die DIN-Adresse K in einem separaten NC-Kanal ausgeführt werden kann.

Ist E direkt programmiert, wird die Funktion zeitsynchronisiert und der Wert von E als Index eines Q-Bits interpretiert. Der Sprung wird nur ausgeführt, wenn das Q-Bit ”1” ist. Im Falle indirekter Programmierung wird der Sprung nur ausgeführt, wenn der Inhalt des angegebenen Parameters ungleich ”0” ist. Eine Zeitsynchronisation erfolgt nur, wenn es sich um einen zeitsynchronisierten Parameter handelt.

Die Bearbeitung des Programms verzweigt in das aufgerufene Unterprogramm L, die Fortsetzung des Hauptprogramms erfolgt nach Rückkehr aus dem Unterprogramm mit dem auf den Unterprogrammaufruf folgenden Satz.

Es dürfen bis zu 7 Unterprogrammaufrufe geschachtelt werden.

Modal wirksame Funktionen sind nach Rückkehr aus einem Unterprogramm weiterhin gültig. Im Unterprogramm ist zunächst der Zustand des Hauptprogramms gültig, dieser kann jedoch geändert werden.

Mit der Angabe einer Satznummer unter der Adresse J ist es möglich, gezielt mit einer bestimmten Satznummer des Unterprogramms zu beginnen. Wenn keine Programmnummer L angegeben ist, erfolgt der Unterprogrammaufruf innerhalb des aktuellen Programms. Die Programmbearbeitung verzweigt dann zu der angegebenen Satznummer. Die Fortsetzung des Programms erfolgt wie bei einem normalen Unterprogrammaufruf nach G99 mit dem der G22 folgenden Satz. Zu beachten ist hierbei, dass die Satznummern immer in numerisch aufsteigender Reihenfolge vergeben werden müssen!

Zusätzlich ist es möglich, unter der Adresse I eine Anzahl von Wiederholungen zu programmieren. Das bedeutet, dass mit dem der G22 folgenden Satz erst begonnen wird, wenn das Programm L I-Mal aufgerufen wurde.

Start eines 2. NC-Kanals

Die Ausführung des Unterprogramms kann auch parallel zum aktuellen Programm erfolgen. Dazu muss bei K die Nummer des NC-Kanals angegeben werden, in dem das Programm gestartet werden soll. Die angegebene Kanalnummer muss größer als die Nummer des aktuellen Kanals (normalerweise 0) und kleiner als die maximale Kanalanzahl in der Maschinenkonstante MK_KANALANZAHL sein.

Ein NC-Kanal kann nur die Achsen ansprechen, die ihm zugeordnet sind. Sollen im Kanal 1 Achsen verfahren werden, so müssen diese beim Start des NC-Kanals mit G22 an den neuen Kanal übergeben werden. Der Kanal 0 kann mit G153 die übergebenen Achsen zurückholen, sobald der Kanal 1 mit M30 beendet wurde.

Das Programm im Kanal 0 kann erst beendet werden, wenn auch das Programm im Kanal 1 beendet ist. Die Synchronisation erfolgt automatisch bei M30.

Beispiel

N120 G22 L1100 I4	Viermaliger Aufruf des Unterprogramms %1100
N270 G22 J4000	Start eines Unterprogramms in der gleichen Programmdatei, welches bei Satznummer N4000 beginnt.
G22 L8000 J500	Aufruf des Unterprogramms %8000 ab Satz 500.
G22 L395 I3 E=P3012	Bedingter Aufruf des Unterprogramms %395 bis zu 3-mal hintereinander. Ein vorzeitiges Ende der Schleife erfolgt, falls das Parameterfeld an der Stelle 3012 den Wert 0 aufweist.
G22 L9001 K1 U0 V0	Start des Programms %9001 im zweiten Kanal mit Übergabe der U- und V-Achse an den neuen NC-Kanal.

3.2.2.12 G24 Positive Verfahrbereichsgrenze definieren

Definition einer Verfahrbereichsgrenze in positiver Verfahrrichtung.

Syntax

G24 AXES

Bedeutung der Adressen

AXES	positive Verfahrbereichsgrenze der angegebenen Achsen

Erläuterung

Sind mit der Wegbedingung Achsenadressen angegeben, so werden die programmierten Werte, bezogen auf den durch den Grundoffset definierten Maschinennullpunkt, ab P304 in das Parameterfeld eingetragen. Ist hingegen keine weitere DIN-Adresse außer der Wegbedingung angegeben, so wird das Parameterfeld wieder mit den in den Maschinenkonstanten voreingestellten Werten besetzt.

Um die mit G24 verschobenen Verfahrbereichsgrenzen zu aktivieren, muss anschließend eine G26 mit den entsprechenden Achsbuchstaben programmiert werden.

Beispiel

N100 G24	Für alle Achsen der Anlage werden die positiven Verfahrbereichsgrenzen auf den durch die Maschinenkonstanten bestimmten Wert vordefiniert.
G24 X2345.554	Die positive Verfahrbereichsgrenze der Achse X wird auf die absolute Position 2345.554 definiert.

3.2.2.13 G25 Negative Verfahrbereichsgrenze definieren

Definition einer Verfahrbereichsgrenze in negativer Verfahrrichtung.

Syntax

G25 AXES

Bedeutung der Adressen

AXES	negative Verfahrbereichsgrenze der angegebenen Achsen

Erläuterung

Sind mit der Wegbedingung Achsenadressen angegeben, so werden die programmierten Werte, bezogen auf den durch den Grundoffset definierten Maschinennullpunkt, ab P336 in das Parameterfeld eingetragen. Ist hingegen keine weitere DIN-Adresse außer die Wegbedingung angegeben, so wird das Parameterfeld wieder mit den in den Maschinenkonstanten voreingestellten Werten besetzt.

Um die mit G25 verschobenen Verfahrbereichsgrenzen zu aktivieren, muss anschließend eine G26 mit den entsprechenden Achsbuchstaben programmiert werden.

Beispiel

N150 G25	Für alle Achsen der Anlage werden die negativen Verfahrbereichsgrenzen auf den durch die Maschinenkonstanten bestimmten Wert vordefiniert.
N200 G25 X-115	Die negative Verfahrbereichsgrenze der Achse X wird auf die absolute Position -115 definiert.

3.2.2.14 G26 Verfahrbereichsgrenzen aktivieren

Aktivieren bzw. deaktivieren von Verfahrbereichsgrenzen.

Syntax

G26 AXES

Bedeutung der Adressen

AXES	Gültigkeitskennung (Wert beliebig)

Erläuterung

Sind neben der Wegbedingung Achsadressen programmiert, werden die im Parameterfeld eingetragenen Werte für die positiven und negativen Verfahrbereichsgrenzen übernommen und damit aktiviert. Sind außer der G26 keine weiteren DIN-Adressen programmiert, werden die Verfahrbereichsgrenzen aller (!) Achsen aufgehoben.

Beispiel

N250 G26 X0	Die positive und die negative Verfahrbereichsgrenze der Achse X wird aktiviert.

3.2.2.15 G27 Sprungfunktion mit Wiederholungszähler

Programmierung einer Schleife, die n-mal durchlaufen werden soll.

Syntax

G27 X Z

Bedeutung der Adressen

X	Sprungziel (Satznummer)
Z	Wiederholungszähler

Erläuterung

Die programmierte Satznummer wird entsprechend der Zahl im Wiederholungszähler Z durchlaufen. Die Ziel-Satznummer muss immer kleiner sein als die Satznummer, in der G27 programmiert ist, d. h. es sind nur Rücksprünge erlaubt. Bei Nichtbeachtung erfolgt ein Abbruch mit einer Fehlermeldung.

Die Ziel-Satznummer muss vorhanden sein, andernfalls erfolgt eine Fehlermeldung bei der Abarbeitung des Programms.

Der maximale Wert, der unter Z programmiert werden kann ist 65535; ist der Wert 0 angegeben, so wird die Schleife 65536-mal durchlaufen.

Beispiel

N10 G27 X10 Z3	Das Programm führt drei mal die Sätze zwischen N10 und der G27 aus, bevor es mit dem Satz nach der G27 fortfährt.

3.2.2.16 G30, G31 Strecken- bzw. Bahnsteuerung

Mit der Wegbedingung G30 wird die Steuerung in den Streckenbetrieb und mit G31 in den Bahnbetrieb geschaltet.

Syntax

G30
G31

Erläuterung

Im Bahnbetrieb fahren alle programmierten Achsen auf einer Bahn, wobei die Geschwindigkeit der Achsen sich aus der programmierten Bahngeschwindigkeit ergibt. Alle Achsen erreichen ihren programmierten Endpunkt gleichzeitig.

Beim Streckenbetrieb fahren die programmierten Achsen ihren Weg zwar gleichzeitig, aber unabhängig voneinander. Alle Achsen fahren mit der für sie gültigen Geschwindigkeit ihren Endpunkt an. Der Endpunkt wird in Abhängigkeit vom programmierten Weg nicht gleichzeitig erreicht.

Die Abarbeitung des nächsten Satzes beginnt, nachdem die letzte Achse ihren Zielpunkt erreicht hat.

Die Wegbedingung G31 (Bahnbetrieb) ist nach dem Einschalten bzw. Rücksetzen der Steuerung voreingestellt, wenn nicht in den Maschinenkonstanten mit MK_VOREINSTELLUNG eine andere Voreinstellung gewählt wurde.

Beispiel

N5 G30	Mit Satz 5 wird die Steuerung in den Streckenbetrieb umgeschaltet.

3.2.2.17 G33 Kopplung zwischen Bahn und Rotationsachse einschalten

Mit der G33 wird die Kopplung zwischen der Bahn bzw. einer an der Bahn beteiligten Achse und eine Rotationsachse eingeschaltet.

Syntax

G33 A B C L E oder
G33 I J K

Bedeutung der Adressen

ABC	Rotationsachsen, die der Bahn nachgeführt werden sollen. (Wert beliebig).
L	Bahnlänge, die während einer ganzen Umdrehung der Rotationsachse zurückgelegt werden soll.
E	Drehzahlvorwahl der Rotationsachse in 1/min. Die resultierende Bahngeschwindigkeit ist F=E*L.
I	Gewindesteigung in X-Richtung. Kopplung zwischen X- und A-Achse aktiv.
J	Gewindesteigung in Y-Richtung. Kopplung zwischen Y- und B-Achse aktiv.
K	Gewindesteigung in Z-Richtung. Kopplung zwischen Z- und C-Achse aktiv.

Erläuterung

Die G33 unterscheidet zwei Varianten anhand der Programmierung.

Die erste Variante dient zur Nachführung von Rotationsachsen zum Vorschub auf der Bahn (z. B. Technologie Nähen). Dabei wird pro zurückgelegten Weg auf der Bahn eine Umdrehung der Rotationsachsen ausgeführt. Die Schrittlänge wird pro Satz so korrigiert, so dass eine ganze Anzahl Umdrehungen der Rotationsachsen gefahren wird.

Rampen und Geschwindigkeiten auf der Bahn werden falls erforderlich so nach unten korrigiert, dass die Rotationsachsen nicht überfordert werden. Bei der zweiten Variante wird die Rotationsachse nicht dem Vorschub auf der Bahn nachgeführt, sondern dem Vorschub einer einzelnen Achse. Diese Variante dient speziell zum Gewindeschleifen.

Die programmierte Steigung wird in P685 abgelegt und die Steigungskorrektur in P686 genullt. Die Summe von P685 und P686 ergibt die effektive Gewindesteigung in mm/Umdrehung. Die Steigungskorrektur kann online durch Veränderung von P686 beeinflusst werden.

Während die Gewindekopplung aktiv ist, kann die beteiligte Rotationsachse nicht unabhängig positioniert werden. Die Kopplung ist modal wirksam und muss mit G34 ausgeschaltet werden. Am Programmende wird die Kopplung der ersten Variante automatisch ausgeschaltet. Beide Varianten können nicht gleichzeitig aktiv sein. Bei der ersten Variante können mehrere Rotationsachsen der Bahn nachgeführt werden, wenn diese gemeinsam bei der G33 programmiert werden.

Beispiel

G33 L5 A0	Pro Umdrehung der A-Achse sollen 5 mm auf der Bahn zurückgelegt werden.
G33 K1.5	Die Gewindesteigung soll 1.5 mm pro Umdrehung betragen. Pro 1.5 mm Vorschub der Z-Achse soll die C-Achse also eine Umdrehung machen.

3.2.2.18 G34 Kopplung zwischen Bahn und Rotationsachse ausschalten

Mit der G34 wird die mit G33 eingeschaltete Bahnkopplung wieder ausgeschaltet.

Syntax

G34

Erläuterung

Die G34 hebt die Kopplung zwischen der Bahn bzw. einer an der Bahn beteiligten Achse und der Rotationsachse wieder auf.

Bei der zweiten Variante der G33, dem Gewindeschleifen, ist dies u. a. notwendig, um die beteiligte Rotationsachse allein zu positionieren.

Beispiel

G34	Kopplung ausschalten. Rotationsachse wird nicht mehr mitgeführt.

3.2.2.19 G36 Modales Oszillieren einschalten

Modales Oszillieren für eine Achse einschalten.

Syntax

G36 AXES I J K F

Bedeutung der Adressen

AXES	Zu oszillierende Achse und Hub in mm
I	Verweilzeit am linken Umkehrpunkt (Default 0)
J	Verweilzeit am rechten Umkehrpunkt (Default 0)
K	Bewertung von I und J nicht programmiert: Keine Verweilzeit (I u. J egal). 0: I, J geben die Verweilzeit in Sekunden an. 1: I, J geben die Anzahl der Werkstückumdrehungen am Umkehrpunkt an. Nicht implementiert.
F	Oszillationsgeschwindigkeit [mm/min]

Erläuterung

Die modale Pendelbewegung mit programmierbarem Hub und Geschwindigkeit wird eingeschaltet. Der Hub bezieht sich auf die Position der Achse zu Beginn des Satzes. An den Umkehrpunkten kann eine Verweilzeit programmiert werden. Die Angabe der Zeiten kann für beide Umkehrpunkte unabhängig voneinander erfolgen. Die Bewegung wird so lange ausgeführt, bis sie durch G37 angehalten wird. Wenn mehr als eine Achse programmiert ist, wird die Oszillationsbewegung für jede Achse unabhängig ausgeführt. Die Angabe der Verweilzeiten und der Oszillationsgeschwindigkeit gilt dann für jede programmierte Achse.

Beispiel

N10 G36 Z40 J1 F100	Die Z-Achse führt eine Pendelbewegung mit einem Hub von 40 mm und einer Geschwindigkeit von 100 mm/min aus. Am rechten Umkehrpunkt erfolgt eine Verweilzeit von 1 sec.

3.2.2.20 G37 Modales Oszillieren ausschalten

Syntax

G37 AXES

Bedeutung der Adressen

AXES	Gültigkeitskennung für die Achsen, für die das modale Oszillieren ausgeschaltet werden soll. Wert = 1.

Erläuterung

Mit der Wegbedingung G37 wird das modale Oszillieren (siehe G36) ausgeschaltet. Um dies zu erreichen, wird in einem Satz mit der Wegbedingung G37 unter der Adresse der Achse, deren modale Oszillation ausgeschaltet werden soll, eine 1 programmiert.

Die Achse wird dann sofort gestoppt, d. h. sie fährt ihren Oszillationshub nicht zu Ende.

Beispiel

N15 G37 Z1	In Satz 15 wird die modale Oszillation der Z-Achse gestoppt.

3.2.2.21 G40 Aufheben der Werkzeugkorrektur

Abschalten der durch G41 / G42 eingeschalteten Werkzeugbahnkorrektur.

Syntax

G40 E

Bedeutung der Adressen

E	Auswahl der Abfahrstrategie, nur bei Standard-WRK. 0: Nach der G40 muss ein Abfahrsatz folgen (Default). 1: Kein Abfahrsatz nach G40, die Achsen bleiben auf der letzten korrigierten Position.

Erläuterung

G40 schaltet das mit G41 bzw. G42 eingeschaltete Korrekturverfahren wieder aus. Standardmäßig ist eine Werkzeugradiuskompensation vorgesehen.

Mit dem nächsten Verfahrsatz wird der nicht korrigierte Zielpunkt angefahren. Es ist zu beachten, dass bei der WRK im Anschluss an G40 immer ein Verfahrsatz programmiert werden muss, wenn nicht E1 angegeben ist. Der Startpunkt dieses Satzes ist auf der kompensierten Bahn, der Zielpunkt des Satzes ist unkompensiert!

Beispiel

N 100 G1 X10	Y10 Letzter Kontursatz innerhalb der WRK
N 110 G40	Ausschalten der WRK
N 120 G1 X0	Abfahrsatz

3.2.2.22 G41 Werkzeugbahnkorrektur - links, G42 Werkzeugbahnkorrektur - rechts

Einschalten der Werkzeugradiuskompensation oder eines applikationsspezifischen Korrekturverfahrens für die Werkzeugbahn.

Syntax

G41 R J L
G42 R J L

Bedeutung der Adressen

R	Zu kompensierender Werkzeugradius
J	Werkzeugorientierung (Wert 0 ... 9)
L	Übergangsart 0: linear, 1: zirkular

Erläuterung

Die Wegbedingungen schalten die Werkzeugradiuskompensation oder ein applikationsspezifisches Korrekturverfahren ein. Der Werkzeugradius wird aus dem Parameterfeld (P700) übernommen, welches bei einer T-Umschaltung vorbesetzt wird oder kann gegebenenfalls mit dem Kennbuchstaben R direkt programmiert werden. Bei applikationsspezifischen Korrekturen werden die Maße über vereinbarte Schnittstellen übernommen (Parameterfeld, Maschinenkonstanten). G41 aktiviert eine Korrektur links, G42 rechts der programmierten Bahn, bezogen auf die Richtung der Werkzeugbewegung. Die Kompensation gilt für die angewählte Ebene (G17, G18, G19). Nach Anwahl der Kompensation wird mit dem nachfolgenden Satz die kompensierte Bahn angefahren. Der Startpunkt dieses Satzes ist die momentane Ist-Position, der Zielpunkt liegt auf der kompensierten Bahn (Schnittpunkt mit nachfolgendem Satz).

Die entsprechende Werkzeugorientierung J kann aus der nachfolgenden Tabelle entnommen werden, die den Zusammenhang zwischen den Koordinaten des Schneidenmittelpunktes (Xs,Ys) und den Koordinaten des Werkzeugschneidenpunktes (Xp,Yp) für jeden Orientierungswert beinhaltet.

J	Xp	Yp
0	Xs	Ys
1	Xs+R	Ys+R
2	Xs-R	Ys+R
3	Xs-R	Ys-R
4	Xs+R	Ys-R
5	Xs+R	Ys
6	Xs	Ys+R
7	Xs-R	Ys
8	Xs	Ys-R

Die Werkzeugbahnkorrektur arbeitet nach dem Schnittpunktverfahren. Bei großem Konturübergangswinkel kann der Schnittpunkt sehr weit von der Bahn entfernt liegen. In dem Fall werden lineare Zwischensätze (L0, Konturübergangswinkel > 270 Grad) oder ein zirkularer Übergang (L1, Konturübergangswinkel > 180 Grad) generiert. Die Kompensation wird abgeschaltet durch G40. Ein Wechsel von G41 <-> G42 ohne dazwischenliegendes G40 ist erlaubt.

Beispiel

N100 G41 R1	Werkzeugradiuskorrektur mit 1 mm Werkzeugradius einschalten.
N110 G1 X10	Anfahrsatz
N120 G1 X20 Y10	Erster Kontursatz

3.2.2.23 G53 Temporäre Koordinatenverschiebung aufheben

Hebt die temporäre Koordinatenverschiebung im aktuellen Koordinatensystem wieder auf.

Syntax

G53

Erläuterung

Mit der Funktion G53 wird eine durch G54 vorgenommene temporäre Koordinatenverschiebung aufgehoben. Es gelten wieder die ursprünglich festgelegten Nullpunkte. Dies gilt nur für das aktuelle Werkstückkoordinatensystem.

Beispiel

G53 S1	Die zuvor vorgenommene temporäre Nullpunktverschiebung in S1 wird rückgängig gemacht.

3.2.2.24 G54 Temporäre Nullpunktverschiebung

Mit der Wegbedingung G54 kann der Nullpunkt des aktuellen Werkstückkoordinatensystems temporär verschoben werden.

Syntax

G54 AXES]

Bedeutung der Adressen

AXES	Achsen, deren Nullpunkt verschoben werden soll.

Erläuterung

Mit G54 kann der Nullpunkt des aktuellen Werkstückkoordinatensystems verschoben werden. Die Verschiebung erfolgt wie bei G92, ist jedoch nur temporär wirksam. Sie wird aufgehoben bei Programmende oder -abbruch oder durch G53.

Zu beachten ist die Sonderstellung von S0. Bei einer Verschiebung von S0 werden alle anderen S-Koordinatensysteme mit verschoben. Ist die Maschinenkonstante MK_S0T0_VERSATZ_ERLAUBT nicht gesetzt, wird die Verschiebung in S0 unterdrückt, in allen anderen Koordinaten wird sie jedoch wirksam.

Beispiel

G54 X10 Y10	Die Nullpunkte der Achsen X und Y im aktuellen Koordinatensystem (S) werden um jeweils 10 mm verschoben.

3.2.2.25 G60 Genauhalt Ein/Aus

Umschalten des Verhaltens der Bahnsteuerung am Satzende.

Syntax

G60 X Y

Bedeutung der Adressen

X	Wahlschalter nicht programmiert: Genauhalt ein. 0: Genauhalt aus, LookAhead ein. >0: Wartezeit am Satzende in Sekunden.
Y	Aktivieren von Genauhalt mit Schleppabstandsüberwachung. Der programmierte Wert hat keine Bedeutung.

Erläuterung

Mit der Funktion G60 kann das Verhalten der Bahnsteuerung am Satzende eines Verfahrsatzes beeinflusst werden. Die Grundeinstellung ist ”Genauhalt ohne Schleppabstandsüberwachung”.

Genauhalt ohne Schleppabstandsüberwachung:

Die Zielposition des Satzes wird über eine Bremsrampe angefahren. Die Größe der Rampe wird in Maschinenkonstanten festgelegt. Die Bahngeschwindigkeit beträgt am Satzende ”0”.

Genauhalt mit Wartezeit:

Die Zielposition wird wie beim Genauhalt angefahren. An dieser Position wird für die unter X programmierte Zeit gewartet, bevor der Satz beendet wird.

Genauhalt mit Schleppabstandsüberwachung:

Nach dem Erreichen der Zielposition wartet die Steuerung, bis der Schleppabstand der beteiligten Achsen innerhalb des konfigurierten Genauhaltfensters (MK_SCHLEPPGENAUHALT) ist. Die maximale Wartezeit kann bei X programmiert werden, voreingestellt sind 10 Sekunden.

Genauhalt aus:

Die Zielposition des Satzes wird mit der programmierten Geschwindigkeit angefahren, wenn ein weiterer Verfahrsatz folgt. Wenn die Geschwindigkeit des nachfolgenden Satzes niedriger ist, wird zum Satzende auf diese Geschwindigkeit gebremst. Die Geschwindigkeit am Satzende wird beeinflusst durch den Konturverlauf, d. h. den Übergangswinkel der beiden Sätze (”look ahead”).

Das Bremsen erfolgt über eine Rampe. Wenn kein weiterer Verfahrsatz folgt, wird die Zielposition wie bei ”Genauhalt ohne Schleppabstandsüberwachung” angefahren.

Beispiel

G60 X0	Ab sofort soll mit ausgeschaltetem Genauhalt (mit ”look ahead”) gefahren werden.

3.2.2.26 G61 Satzvorverarbeitung anhalten

Zeitsynchronisation mit Grobinterpolator durchführen.

Syntax

G61

Erläuterung

Mit der Funktion G61 kann die Satzvorverarbeitung des Interpreters vorübergehend angehalten werden, d. h. die Interpretation des nächsten Satzes wird angehalten bis der letzte Satz im Auftragspuffer des Grobinterpolators abgearbeitet wurde.

Wenn die G61 innerhalb einer Kontur verwendet wird, kommt es dabei zwangsläufig zum kurzzeitigen Stillstand der Achsen auf der Bahn, da ohne Anschlusssatz keine Anschlussgeschwindigkeit bestimmt werden kann.

Beispiel

N10 G1 X10 Z20	Satz innerhalb der Schleife
...	Weitere Sätze in der Schleife
N50 G61	Satzvorverarbeitung anhalten, bis der Satz vor N50 ausgeführt wurde.
N60 :P1999=P1999+1	Parameter zur Anzeige der Schleifendurchläufe erhöhen.
N70 G27 X10 Z33	Schleife 33-mal durchlaufen.

3.2.2.27 G74 Referenzpunktfahrt

Mit der Wegbedingung G74 wird die Referenzpunktfahrt ausgeführt.

Syntax

G74 AXES

Bedeutung der Adressen

AXES	Gültigkeitskennung (Wert beliebig)

Erläuterung

Es gibt zwei Möglichkeiten der Referenzpunktfahrt.

Falls in einem DIN-Satz die Funktion G74 ohne weitere Adressen programmiert ist, wird die automatische Referenzpunktfahrt in der in den Maschinenkonstanten festgelegten Reihenfolge ausgeführt.

Bei Angabe einzelner Achsen wird die Referenzpunktfahrt nur in den programmierten Achsen durchgeführt.

Die Richtung der Referenzpunktfahrt ist durch die entsprechende Maschinenkonstante der Achse vorgegeben, d. h. die Richtung entspricht der automatischen Referenzpunktfahrt. Eine Achse wird programmiert durch Eingabe einer beliebigen einstelligen Zahl unter der zugehörigen Achsadresse.

Beispiel

G74 X	Die Referenzpunktfahrt wird für die konfigurierten Achsen in der konfigurierten Reihenfolge durchgeführt.

3.2.2.28 G75 Maßstabsfaktoren für Eingabeeinheiten

Setzen des Maßstabsfaktors der einzelnen Achsen für die programmierten Eingabeeinheiten.

Syntax

G75 AXES

Bedeutung der Adressen

AXES	Angabe der Achse und des Faktors

Erläuterung

Die G75 bewirkt eine Streckung oder Stauchung der programmierten Kontur vor der Verarbeitung durch interne Korrekturmodule. Diese kann unsymmetrisch sein, wenn unterschiedliche Skalierungsfaktoren für die einzelnen Achsen programmiert werden. Dabei ist zu beachten, dass nur die jeweils programmierten Zielkoordinaten, Mittelpunktkoordinaten und Radien skaliert werden, d. h. Kreise bleiben weiterhin Kreise und werden niemals zu Ellipsen expandiert.

Ein negativer Skalierungsfaktor bewirkt die Spiegelung um die jeweilige Achse. Die Programmierung der Funktion ohne Angabe von Achsen setzt alle Faktoren wieder auf ”1”.

Beispiel

G75 X2 Y-3	Skalierungsfaktor 2 für X und -3 für Y
G1 X10 Y5	Fahren auf X20 und Y-15

3.2.2.29 G76 Maßstabsfaktoren für Impulsbewertung

Setzen des Maßstabsfaktors für die Impulsbewertung der einzelnen Achsen.

Syntax

G76 AXES

Bedeutung der Adressen

AXES	Angabe der Achse und des Faktors

Erläuterung

Mit der G76 kann die Impulsbewertung (Impulse pro Weg) der Achsen geändert werden. Dies entspricht einer linearen Transformation, wie sie z. B. durch eine Umschaltung der Getriebestufe nötig wird.

Die G76 bewirkt wie die G75 eine Streckung oder Stauchung der programmierten Kontur. Im Gegensatz zur G75 werden jedoch auch die Geschwindigkeiten und Beschleunigungen beeinflusst. Die max. zulässigen Geschwindigkeiten und Beschleunigungen aus den Maschinenkonstanten werden dazu entsprechend angepasst. Durch die Änderung der Impulsbewertung ergeben sich bei unsymmetrischen Skalierungsfaktoren geometrisch andere Konturen als programmiert, z. B. werden Kreise zu Ellipsen und die Geschwindigkeit auf der Bahn wird nicht mehr konstant sein.

Ein negativer Skalierungsfaktor bewirkt die Spiegelung um die jeweilige Achse. Die Programmierung der Funktion ohne Angabe von Achsen setzt alle Faktoren wieder auf ”1”.

Beispiel

G76 C-1	Bewegungssinn der C-Achse invertieren.

3.2.2.30 G88 Grunddrehung

Drehung der Systeme S1 - S31 in der XY-Ebene.

Syntax

G88 C X Y

Bedeutung der Adressen

C	Grunddrehwinkel in Grad.
X, Y	Bestimmung des Grunddrehwinkels aus Delta-X und Delta-Y

Erläuterung

Die Grunddrehung dient der Definition der Lage eines Werkstückes (S1 ... S31) relativ zum Maschinenarbeitsraum (S0). Die Drehung wirkt für S1 ... S31 gemeinsam um den gleichen Winkel in der XY-Ebene. Im Referenz-Werkstückkoordinatensystem S0 ist die Drehung nicht wirksam, gleichwohl kann die Drehung aber dort, stehend in S0, definiert werden. Sie wird dann erst bei Umschaltung nach S1 ... S31 aktiv.

Bei der Einschaltung der Drehung werden die Ist-Positionen der X, Y und C Achse an die neue Lage angepasst; eine Ausgleichsbewegung findet nicht statt.

Die Angabe des Winkels ist eine Absolutangabe.

Die Drehung wird deaktiviert durch Programmierung der G88 ohne Parameter bzw. durch Angabe des Winkels 0°.

Die Funktion ist modal wirksam und bleibt bis zur Aufhebung oder Rücksetzen der Steuerung aktiv.

Beispiel

N1 G0 X10 Y15 S0 T0 N2 G193 X-10 Y50 S1 N3 G88 C30	Mit Satz 3 wird eine Drehung der XY-Ebene für alle Systeme S1 ... S31 um 30 Grad, relativ zum ungedrehten S0 aktiviert. Die Drehung wird sofort wirksam, da sie in S1 stehend aktiviert wurde.

3.2.2.31 G89 Konturdrehung

Drehung der Systeme S1 ... S31 in der XYZ-Ebene wirkt wie die G88, ist jedoch nur innerhalb eines Programms und additiv zur G88 wirksam.

Syntax

G89 A B C

Bedeutung der Adressen

A	Relativer Drehwinkel der YZ-Ebene in Grad
B	Relativer Drehwinkel der ZX-Ebene in Grad
C	Relativer Drehwinkel der XY-Ebene in Grad

Erläuterung

Die Konturdrehung dient der Definition der Lage einer Konturbeschreibung auf einem Werkstück. Die Drehung wirkt für S1 ... S31 gemeinsam. Im Referenz-Werkstückkoordinatensystem S0 ist die Drehung nicht wirksam, gleichwohl kann die Drehung aber dort, stehend in S0, definiert werden. Sie wird dann erst bei Umschaltung nach S1 ... S31 aktiv.

Bei der Einschaltung der Drehung werden die Ist-Positionen der X, Y, Z, A, B und C Achse an die neue Ebenenlage angepasst; eine Ausgleichsbewegung findet nicht statt.

Die Transformations-Reihenfolge ist immer A->B->C.

Die Angaben der Winkel sind Relativangaben.

Die Drehung wird deaktiviert durch Weglassen aller Parameter und sie wird automatisch am Programmende oder bei Programmabbruch ausgeschaltet. Bei der Deaktivierung wird der Winkel auf 0 zurückgesetzt.

Beispiel

N1 G0 X10 Y15 S0 T0	Mit Satz 4 wird eine Konturdrehung der YZ-Ebene um 10° und der XY-Ebene um 15° für alle Systeme S1 ... S31 aktiviert. Die Drehung wird erst in Satz 5 wirksam, da sie in S0 stehend aktiviert wurde.
N2 G193 X0 Y0 S1	
N3 S0	
N4 G89 A10 C15	Die aktuellen Ist-Positionen werden in Satz 5 auf das gedrehte System umgerechnet.
N5 S1	

3.2.2.32 G90 Absolute Maßangaben (Bezugsmaß)

Umschaltung auf das Bezugsmaßsystem.

Syntax

G90 AXIS R D

Bedeutung der Adressen

AXES	Gültigkeitskennung der Achsen (Wert beliebig)
R	Gültigkeitskennung für Polarradius
D	Gültigkeitskennung für Polarwinkel

Erläuterung

Bei der Bezugsmaßprogrammierung bezieht sich der programmierte Endpunkt der Werkzeugbahn immer auf den Nullpunkt des gewählten Koordinatensystems.

Es ist möglich, einzelne oder alle Achsen auf Bezugsmaßprogrammierung umzuschalten. Bei Angabe einer Achse wird nur diese auf Kettenmaß umgeschaltet; für die anderen bleibt die momentane Bemaßungsart gültig. Wird die G90 ohne DIN-Adressen programmiert, werden alle Achsen ins Bezugsmaßsystem umgeschaltet.

Die Wegbedingung G90 ist modal wirksam und wird erst durch die Wegbedingung G91 (Kettenmaßsystem) ausgeschaltet.

Beispiel

N3 G90	Alle Achsen in das Bezugsmaßsystem umschalten.

3.2.2.33 G91 Inkrementale Maßangaben (Kettenmaßprogrammierung)

Mit der Wegbedingung G91 wird vom Bezugsmaßsystem auf das Kettenmaßsystem umgeschaltet. Dies kann generell für alle Achsen (ohne Adressparameter) und auch für einzelne Achsen gemacht werden.

Syntax

G91 AXES R D

Bedeutung der Adressen

AXES	Gültigkeitskennung der Achsen (Wert beliebig)
R	Gültigkeitskennung für Polarradius
D	Gültigkeitskennung für Polarwinkel

Erläuterung

Im Kettenmaßsystem wird der Endpunkt der programmierten Achse immer relativ zu ihrem Startpunkt programmiert. Das Vorzeichen gibt die Richtung der Achsbewegung an; der Startpunkt ist die momentane Achsposition.

Es ist möglich, einzelne oder alle Achsen auf Kettenmaßprogrammierung umzuschalten. Bei Angabe einer Achse wird nur diese auf Kettenmaß umgeschaltet; für die anderen bleibt die momentane Bemaßungsart gültig. Wird die G91 ohne DIN-Adressen programmiert, werden alle Achsen ins Kettenmaßsystem umgeschaltet.

Die Wegbedingung G91 ist modal wirksam und wird erst durch die Wegbedingung G90 (Bezugsmaßsystem) ausgeschaltet.

Beispiel

N7 G91	Alle Achsen ins Kettenmaßsystem umgeschalten.

3.2.2.34 G92 Relative Nullpunktverschiebung

Mit der Wegbedingung G92 kann der Nullpunkt des aktuellen Koordinatensystems incremental verschoben werden.

Syntax

G92 AXES

Bedeutung der Adressen

AXES	Achsen, deren Nullpunkt verschoben werden soll.

Erläuterung

Die incrementale Nullpunktverschiebung wird mit der Wegbedingung G92 und den Adressbuchstaben der Achsen programmiert, bei denen der Nullpunkt verschoben werden soll.

Unter den Adressbuchstaben der Achsen wird der Wert programmiert, um den der Nullpunkt der Achse verschoben werden soll. Mit der Wegbedingung G92 können die Nullpunkte aller gewünschten Achsen in einem Satz verschoben werden.

Wenn im gleichen Satz ein T programmiert ist, erfolgt die Verschiebung im aktuellen Werkzeugkoordinatensystem (T0 ... T31), sonst wird das aktuelle Werkstückkoordinatensystem (S0 ... S31) verschoben.

Zu beachten ist die Sonderstellung von S0. Bei einer Verschiebung von S0 werden alle anderen S-Koordinatensysteme mit verschoben. Ist die Maschinenkonstante MK_S0T0_VERSATZ_ERLAUBT nicht gesetzt, wird beim Versuch S0 zu verschieben, eine Fehlermeldung generiert und das Programm abgebrochen.

Beispiel

N20 G92 X0.03 T1	Hier wird der Nullpunkt der X-Achse in T1 um +30 mm verschoben.

3.2.2.35 G93 Absolute Nullpunktverschiebung

Mit G93 kann der Nullpunkt des aktuellen Koordinatensystems auf einen absolut programmierten Wert verschoben werden.

Syntax

G93 AXES

Bedeutung der Adressen

AXES	Achsen, deren Nullpunkt verschoben werden soll

Erläuterung

Die absolute Nullpunktverschiebung wird mit der Wegbedingung G93 und den Adressbuchstaben der Achsen programmiert, bei denen der Nullpunkt verschoben werden soll.

Unter den Adressbuchstaben der Achsen wird direkt der neue Nullpunktoffset des zu verschiebenden Koordinatensystems programmiert. Mit der Wegbedingung G93 können die Nullpunkte aller gewünschten Achsen in einem Satz verschoben werden.

Wenn im gleichen Satz ein T programmiert ist, erfolgt die Verschiebung im aktuellen Werkzeugkoordinatensystem (T0 ... T31), sonst wird das aktuelle Werkstückkoordinatensystem (S0 ... S31) verschoben.

Zu beachten ist die Sonderstellung von S0. Bei einer Verschiebung von S0 werden alle anderen S-Koordinatensysteme mit verschoben. Ist die Maschinenkonstante MK_S0T0_VERSATZ_ERLAUBT nicht gesetzt, wird beim Versuch S0 zu verschieben, eine Fehlermeldung generiert und das Programm abgebrochen.

Beispiel

N20 G93 Z50 T1	Hier wird der Nullpunktoffset der Z-Achse (Werkzeuglänge) in T1 auf 50 mm gesetzt.

3.2.2.36 G99 Rücksprung

Unterprogrammende mit Rücksprung zum rufenden Programm.

Syntax

G99

Erläuterung

Die Wegbedingung G99 muss in jedem Unterprogramm als letzter Befehl stehen.

Die Steuerung setzt das Programm im Hauptprogramm mit dem Satz fort, der nach dem aufrufenden Satz programmiert wurde.

In einem Satz mit der Wegbedingung G99 darf keine andere Adresse programmiert werden.

Modal wirksame Funktionen sind nur in dem Unterprogramm gültig, in dem sie gesetzt wurden. Nach Beendigung des Unterprogramms sind sie nicht mehr gültig.

Beispiel

N5 G99	Unterprogramme beenden und im aufrufenden Programm weiterarbeiten.

3.2.2.37 G100 Polarkoordinaten: Geradeninterpolation im Eilgang

Die Wegbedingung G100 entspricht funktional der Wegbedingung G00 bei der Programmierung der Zielpositionen im rechtwinkligen Koordinatensystem.

Syntax

G100 AXES U W

Bedeutung der Adressen

AXES	Mittelpunktkoordinaten
U	Polarradius
W	Polarwinkel

Erläuterung

Der Mittelpunkt des Polarkoordinatensystems wird jeweils durch zwei Adressen bestimmt: X/Y bei G17, X/Z bei G18, Y/Z bei G19. Die Zuordnung der Hauptachsen kann mit G16 geändert werden. Bei Verwendung der Achsen U und W als Hauptachsen, können diese nicht für die Mittelpunktprogrammierung verwendet werden.

Die Mittelpunktkoordinaten können im Absolutmaß oder im Kettenmaß programmiert werden.

Die Zielposition wird durch den Polarradius U und den Polarwinkel W im Bezug auf den aktuellen Mittelpunkt bestimmt.

Der Radius und der Winkel können im Absolutmaß oder im Kettenmaß programmiert werden.

Die programmierten Mittelpunktkoordinaten sind modal wirksam.

Die Wegbedingung G100 ist modal wirksam.

Beispiel

N100 G17 N110 G0 X50 Y0 N120 G100 X10 Y10 U20 W45	Hier wird der Nullpunktoffset der Z-Achse (Werkzeuglänge) in T1 auf 50 mm gesetzt.

3.2.2.38 G101 Polarkoordinaten: Geradeninterpolation

Die Wegbedingung G101 entspricht funktional der Wegbedingung G01 bei der Programmierung der Zielpositionen im rechtwinkligen Koordinatensystem.

Syntax

G101 AXES U W F E L

Bedeutung der Adressen

AXES	Mittelpunktkoordinaten
U	Polarradius
W	Polarwinkel
F	Bahngeschwindigkeit
E, L	Auswahl der Vorschubgeschwindigkeit über Drehzahl (E) und Schrittweite (L). F=E*L.

Erläuterung

Der Mittelpunkt des Polarkoordinatensystems wird jeweils durch zwei Adressen bestimmt: X/Y bei G17, X/Z bei G18, Y/Z bei G19. Die Zuordnung der Hauptachsen kann mit G16 geändert werden. Bei Verwendung der Achsen U und W als Hauptachsen, können diese nicht für die Mittelpunktprogrammierung verwendet werden.

Die Mittelpunktkoordinaten können im Absolutmaß oder im Kettenmaß programmiert werden.

Die Zielposition wird durch den Polarradius U und den Polarwinkel W im Bezug auf den aktuellen Mittelpunkt bestimmt.

Der Radius und der Winkel können im Absolutmaß oder im Kettenmaß programmiert werden.

Die programmierten Mittelpunktkoordinaten sind modal wirksam.

Die Wegbedingung G101 ist modal wirksam.

Beispiel

```
N200 G18
N210 G0 X0 Z20
N220 G101 X10 Z10 U30 W45 F1000
```

3.2.2.39 G102/G103 Polarkoordinaten: Kreisinterpolation

Die Wegbedingung G102/G103 entspricht funktional der Wegbedingung G02/G03 bei der Programmierung des Kreises mit Mittelpunktkoordinaten im rechtwinkligen Koordinatensystem.

Syntax

```
G102 AXES  W  F  E  L
G103 AXES  W  F  E  L
```

Bedeutung der Adressen

AXES	Mittelpunktkoordinaten
W	Polarwinkel
F	Bahngeschwindigkeit
E, L	Auswahl der Vorschubgeschwindigkeit über Drehzahl (E) und Schrittweite (L). F=E*L.

Erläuterung

Der Mittelpunkt des Polarkoordinatensystems wird jeweils durch zwei Adressen bestimmt: X/Y bei G17, X/Z bei G18, Y/Z bei G19. Die Zuordnung der Hauptachsen kann mit G16 geändert werden. Bei Verwendung der Achsen U und W als Hauptachsen, können diese nicht für die Mittelpunktprogrammierung verwendet werden.

Die Mittelpunktkoordinaten können im Absolutmaß oder im Kettenmaß programmiert werden. Der Mittelpunkt des Polarkoordinatensystems bestimmt gleichzeitig den Mittelpunkt des Kreises, daher ist die Programmierung des Kreisradius nicht notwendig.

Die Zielposition wird durch den Polarwinkel W im Bezug auf den aktuellen Mittelpunkt bestimmt. Der Winkel kann im Absolutmaß oder im Kettenmaß programmiert werden.

Die programmierten Mittelpunktkoordinaten sind modal wirksam.

Die Wegbedingung G102/G103 ist modal wirksam.

Beispiel

```
N100 G17
N110 G01 X50 Y0 F5000
N120 G103 X0 Y0 W180
N130 G102 X10 Y10 W0
```

3.2.2.40 G110 Polarkoordinaten: Mittelpunkt übernehmen

Die erreichte Sollposition als neuen Mittelpunkt im Polarkoordinatensystem übernehmen.

Syntax

G110 U W

Bedeutung der Adressen

U	Polarradius
W	Polarwinkel

Erläuterung

Die aktuellen Positionen der Hauptachsen werden als neue Mittelpunktkoordinaten übernommen und die mit dem Radius U und dem Winkel W programmierte Zielposition von diesem neuen Mittelpunkt aus mit G101 angefahren.

Wenn weder der Radius U noch der Winkel W programmiert werden, wird keine Bewegung ausgeführt.

Der Radius und der Winkel können im Absolutmaß oder im Kettenmaß programmiert werden.

Beispiel

```
N200 G17
N210 G0 X10 Y10
N220 G102 X20 W0 F1000
N230 G110 U20
N240 G102 W180
```

3.2.2.41 G111 Polarkoordinaten: Mittelpunktprogrammierung mit Winkel und Radius

Syntax

G111 U W

Bedeutung der Adressen

U	Polarradius
W	Polarwinkel

Erläuterung

Der Polarradius U und der Polarwinkel W bestimmen den neuen Mittelpunkt des Polarkoordinatensystems. Es wird keine Achsbewegung ausgeführt.

Der Radius und der Winkel können im Absolutmaß oder im Kettenmaß programmiert werden.

Beispiel

```
N10 G18
N20 G0 X0 Z50
N30 G101 X0 Z50 U50 W90 F2000
N40 G111 U50 W180
N50 G103 W90
```

3.2.2.42 G112 Tangentiale Nachführung ein

Die tangentiale Nachführung der C-Achse wird eingeschaltet.

Syntax

G112 AXES

Bedeutung der Adressen

AXES	Gültigkeitskennung und Mode für nachzuführende Rotationsachse. Es darf nur eine Achse angegeben sein!

Erläuterung

Für Bahnkonturen mit tangentialen bzw. annähernd tangentialen Übergängen ist eine automatische tangentiale Nachführung einer Rotationsachse möglich. Hierbei gibt es zwei Modi, die als Zahlenwert unter dem jeweiligen Achskennbuchstaben programmiert werden müssen.

Mode 1: Tangentiale Nachführung ab momentaner Position der Rotationsachse (Standard)

Mode 2: Tangentiale Nachführung mit schlagartiger Zustellung der Rotationsachse am Beginn der Interpolationsaufgabe auf den in der Geometriefunktion programmierten Winkel

Die reine Nachführung wirkt nur auf Kreiskonturen; die schlagartige Zustellung wirkt in allen Interpolationsarten.

Hinweis!

Diese Funktion errechnet nicht den aktuellen Bahnvektor bei Geradeninterpolation, d. h. es findet keine automatische Ausrichtung der Rotationsachse auf die aktuelle Bahnorientierung statt.

Die Nachführachse wird geschwindigkeitsmäßig nachgeführt. Die Nachführung erfolgt bis zur maximal zulässigen Achsgeschwindigkeit (MK). Wird diese Geschwindigkeit überschritten, wird die Bahngeschwindigkeit entsprechend reduziert.

Die Funktion ist modal wirksam und kann durch G113 abgewählt werden.

Beispiel

G112 C1	Die tangentiale Nachführung für die C-Achse wird im normal Mode eingeschaltet (reine Nachführung ohne Ausrichtung auf programmierten Winkel).

3.2.2.43 G113 Tangentiale Nachführung aus

Ausschalten der tangentialen Nachführung

Syntax

G113 AXES

Bedeutung der Adressen

AXES	Achse, für die tangentiale Nachführung ausgeschaltet werden soll.

Erläuterung

Die mit G112 eingeschaltete tangentiale Nachführung wird für die programmierte Achse ausgeschaltet. Ist keine programmiert, wird die tangentiale Nachführung für alle Achsen ausgeschaltet.

Beispiel

G113	Die tangentiale Nachführung wird ausgeschaltet.

3.2.2.44 G114 6-Achsen-Transformation

Ein- bzw. Ausschalten der 6-Achsen-Transformation zur Bewegung der Werkzeugspitze über 6-Achsen im Raum mit konstanter Geschwindigkeit.

Syntax

G114 I J K A B C E L

Bedeutung der Adressen

I	Werkzeuglänge längs der X-Achse bei Null-Lage der A-, B- und C-Achse. Entspricht dem Abstand zwischen den Drehpunkten der C-Achse und der A-Achse. (Default 0).
J	Werkzeuglänge längs der Y-Achse bei Null-Lage der A-, B- und C-Achse. Entspricht dem Abstand zwischen den Drehpunkten der A-Achse und der B-Achse. (Default 0).
K	Werkzeuglänge längs der Z-Achse bei Null-Lage der A-, B- und C-Achse. Entspricht dem Abstand zwischen dem Drehpunkt der B-Achse und der Werkzeugspitze. (Default 0).
ABC	Kennung für die Drehachsen, die an der Transformation beteiligt werden sollen. Mindestens eine ist notwendig. Wert sollte 0 sein.
E	Kennung: 0: Werkzeugbahn wird programmiert (Default) 1: Werkstückbahn wird programmiert
L	Kennung: 0: Kinematik mit verschobenen Drehpunkten (Default) 1: Kinematik mit gemeinsamen Drehpunkt der A-, B- und C-Achse

Erläuterung

Mit der 6-Achsen-Transformation lassen sich unterschiedliche Maschinenkinematiken und Werkzeuggeometrien berücksichtigen, z. B. räumliche Werkzeugkorrektur, Werkzeugorientierung und Interpolieren einer Werkstückbahn mit konstanter Bahngeschwindigkeit. Dabei kann sowohl die Bahn der Werkzeugspitze (Werkstückbahn), als auch die Bahn der kartesischen Achsen (Werkzeugbahn) programmiert werden.

Die Transformation wird ausgeschaltet durch Programmierung der G114 ohne Angabe von Parametern und bei Programmende oder Abbruch.

Die Reihenfolge der Transformation ist immer C -> A -> B.

Beispiel

G114 A0 J30 E1	Transformation einschalten, Werkzeuglänge längs Y ist 30 mm

3.2.2.45 G115 Mantel-Transformation

Ein- bzw. Ausschalten der Mantel-Transformation.

Syntax

G115 X Y Z A B C I

Bedeutung der Adressen

XYZ	Auswahl von Quer- und Längsachse. 0: Querachse ≠0: Steigungsfaktor der Längsachse (falls programmiert)
ABC	Auswahl der beteiligten Drehachse und Radius am Konusanfang (bei Position der Längsachse=0)
I	ohne programmierte Achsen: 0: Transformation temporär ausschalten 1: Transformation mit alten Parametern wiedereinschalten

Erläuterung

Bei der Mantel-Transformation werden kartesische Koordinaten auf einer Konusoberfläche abgebildet. Der Konus wird über Radius und Steigung bestimmt. Die kartesischen Koordinaten werden bei der Achsausgabe in Koordinaten für eine Drehachse und eine Längsachse transformiert. Die Drehachse dreht das Werkstück und die Längsachse bewegt das Werkzeug längs der Drehachse. Die dritte mögliche Achse, senkrecht zum Werkstück, wird nicht transformiert.

Um die Transformation auszuschalten, muss die G115 ohne Parameter programmiert werden.

Bei eingeschalteter Transformation kann die beteiligte Rotationsachse nur noch über Programmierung der Querachse bewegt werden.

Beispiel

G115 X0 A50	Einschalten der Mantel-Transformation mit X als Querachse, A als Drehachse und einer beliebigen Längsachse. Der Radius des Zylinders, auf dessen Oberfläche die Kontur abgebildet wird, beträgt 50 mm.

3.2.2.46 G116 Rotationsachsen-Transformation

Transformiert ein virtuelles Achssystem mit den Rotationsachsen B und C in ein physikalisches Achssystem mit den Rotationsachsen A und B‘.

Syntax

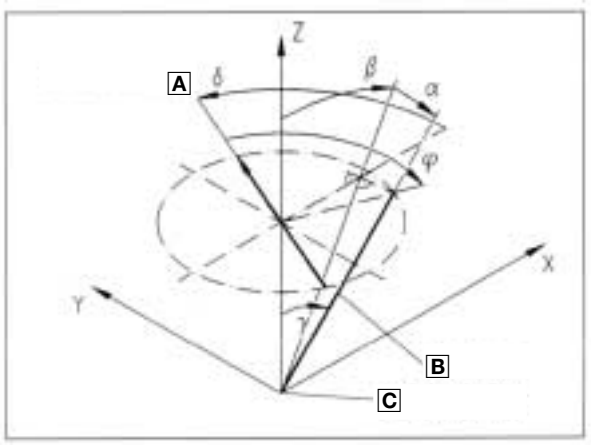
G116 C

Bedeutung der Adressen

C	Verschiebungswinkel φ zwischen Bahnorientierung und Werkzeugorientierung.

Erläuterung

Die Funktion erzeugt die virtuellen Achsen B und C und transformiert deren Winkel γ und δ unter Berücksichtigung des Verschiebungswinkels φ in die Winkel α und β der physikalischen Achsen A und B‘. Dabei gilt folgender kinematischer Zusammenhang:



A Schneidrichtung
B Werkzeug
C Werkzeugspitze

Zum Einschalten der Transformation müssen die Achsen A und B konfiguriert sein. Solange die Transformation eingeschaltet ist, werden die physikalischen Achsen A und B‘ durch die virtuellen Achsen B und C ersetzt und die Programmierung der entsprechenden physikalischen Achsen gesperrt.

Der Definitionsbereich der Winkel α, β und γ liegt zwischen –90° und +90°.

Die Transformation wird durch Programmierung von G116 ohne Parameter ausgeschaltet. Am Programmende oder bei Programmabbruch wird die Transformation automatisch ausgeschaltet.

Beispiel

Das folgende Beispiel soll den Einsatz der G116 am Beispiel einer Schneidtechnologie verdeutlichen. Dazu muss MK_KUNDE = ”SCHNEIDEN” gesetzt werden. Die G41 schaltet in diesem Fall die tangentiale Nachführung des Werkzeugs ein. Nach Einschalten der Transformation wird mit der B-Achse der Gärungswinkel des Schneidwerkzeugs eingestellt.

...	Prolog
G60 X0	Look-Ahead ein
G17	X/Y-Ebene einschalten
G116 C90	Transformation einschalten mit Verschiebungswinkel φ = 90°
G41	Korrekturmodul für tangentiale Nachführung der C-Achse einschalten
G0 X100 Y50 B15 M14	Leerfahrt zum Anfang der Kontur, Schwenken des Werkzeugs um 15°
G3 X50 Y100 R50 F7000	Erster Kontursatz
G1 X-50	Weitere Kontursätze
G3 X-50 Y0 R50	
G1 X50	
G3 X100 Y50 R50 M15	Letzter Kontursatz, M15 = Leerfahrt ein
...	Noch mehr Konturen
G40	Korrekturmodul aus
G116	Transformation aus
...	Epilog

3.2.2.47 G120 Absolute Koordinatenverschiebung über modalen Offset

Setzen der NC-Ist-Position auf den programmierten Wert über Verschiebung des modalen Offsets.

Syntax

G120 AXES

Bedeutung der Adressen

AXES	Achsen, deren Positionen gesetzt werden sollen.

Erläuterung

Mit G120 wird der modale Offset so berechnet, dass die daraus resultierende Ist-Position gleich dem bei der Achsadresse programmierten Wert ist.

Wirkt wie Koordinatenverschiebung mit G193, jedoch auf den modalen Offset. Der modale Offset kann mit G121 genullt werden.

Beispiel

N10 G90	In Satz 30 wird die interne Ist-Position auf 30 und der modale Offset auf 70 gesetzt.
N20 G0 X100	
N30 G120 X30	Die angezeigte Ist-Position (Ist-Pos. + Mod-Pos.) verändert sich dabei nicht.
N40 G0 X130	In Satz 40 wird auf 200=130+70 gefahren. Die angezeigte Ist-Position ist 200.

3.2.2.48 G121 Programmieren des modalen Offsets

Ändern des modalen Offsets mit gleichzeitiger Übernahme des Änderungsbetrags in die NC-Ist-Position.

Syntax

G121 AXES

Bedeutung der Adressen

AXES	Gültigkeitskennung und neuer modaler Offset

Erläuterung

Beim Fahren über die Verfahrtasten (modales Fahren oder Zielpunktfahrt über SPS-Achsen) werden die Achsen über den modalen Offset positioniert. Diese Verfahrbewegung kann prinzipiell zu der Verfahrung im NC-Programm überlagert werden, siehe G122. Die Sollposition in Bezug auf das aktuelle Koordinatensystem der Steuerung ergibt sich aus der Summe von NC-Ist-Position und modalem Offset. Der modale Offset wirkt dabei wie eine Koordinatenverschiebung.

Mit der G121 kann der modale Offset programmiert werden, im einfachsten Fall um ihn zu Nullen und die über die Verfahrtasten ausgelöste Verschiebung aufzuheben. Dabei wird der modale Offset in die NC-Ist-Position übernommen und der modale Offset genullt.

Der modale Offset kann allerdings auch gezielt verwendet werden, um asynchron zum laufenden Programm eine Koordinatenverschiebung herbeizuführen, z. B. in einem Interrupt-Programm (siehe G130 und G150). Bitte beachten Sie, dass eine asynchrone Verschiebung über die Werkzeug- und Werkstückkoordinatensysteme nicht möglich ist!

Beispiel

G121 X0 Y0	Der modale Offset der X- und Y-Achse (P160,P161) wird in die NC-Ist-Position (P0,P1) übernommen und der modale Offset genullt.
G121 X50 Y10	Der modale Offset der X-Achse wird auf 50 und der Y-Achse auf 10 gesetzt, die NC-Ist-Position wird entsprechend verkleinert.

3.2.2.49 G122 Konfigurieren der Wirkung der Verfahrtasten

Syntax

G122 X

Bedeutung der Adressen

X	Verfahrtastenmode

Erläuterung

Mit G122 wird die Wirkung der Verfahrtasten auf die Achsenbewegung konfiguriert. Der Wert bei X hat folgende Bedeutung:

Mode 0: (Default) Die Verfahrtasten sind nur im Handbetrieb und nicht im Automatikprogramm wirksam. Beim Start eines Programms oder eines Einzelsatzes wird der modale Offset automatisch, wie bei G121 beschrieben, genullt.

Mode 1: In diesem Modus wirken die Verfahrtasten als modaler Offset zur programmierten Ist-Position. Sie wirken wie eine Koordinatenverschiebung. Die Verfahrbewegung erfolgt gegebenenfalls überlagert zu einer programmierten Bewegung und kann sowohl im Handbetrieb als auch im Automatikbetrieb erfolgen. Der mit den Verfahrtasten eingestellte Offset kann mit G121 in die Ist-Position übernommen werden.

Mode 2: Im Handbetrieb sind die Verfahrtasten normal wirksam. Der entstehende modale Offset wird beim Start eines Programms oder eines Einzelsatzes automatisch genullt. Im Automatikbetrieb können die Achsen mit den Verfahrtasten entlang einer programmierten Bewegung verfahren werden. Dabei wird die programmierte Geschwindigkeit durch die mit der Verfahrtaste eingestellte Geschwindigkeit ersetzt. Dies gilt auch bei fehlender Vorschubfreigabe! Wenn die Geschwindigkeit negativ ist, erfolgt die Bewegung rückwärts auf der programmierten Kontur.

Im Bahnbetrieb wird die Verfahrtaste der ersten Achse (Achse 0) verwendet. Dabei kann sowohl vorwärts als auch rückwärts über die Satzgrenzen hinweg gefahren werden. Die Anzahl der Sätze über die zurückgefahren werden kann, ist begrenzt (MK_LAH_RUECKLAUFGRENZE). Im Streckenbetrieb wirken die Verfahrtasten der im aktuellen Satz programmierten Achsen. Dabei kann nur vorwärts über die Satzgrenzen hinweg gefahren werden und nur dann, wenn alle Achsen ihren Zielpunkt erreicht haben.

Mode 3: Dies ist eine Kombination aus Mode 1 und 2. Die Verfahrtasten sind sowohl im Handbetrieb als auch im Automatikbetrieb wirksam. Der modale Offset wird nicht automatisch genullt. Bei fehlender Vorschubfreigabe kann wie im Mode 3 auf der programmierten Kontur vor- und zurückgefahren werden.

Mode 4: Die Verfahrtasten sind im Handbetrieb und im Automatikbetrieb wirksam. Der modale Offset wird jedoch bei der nächsten Gelegenheit automatisch genullt. Die Vorwärts- oder Rückwärtsbewegung auf der programmierten Kontur ist nicht möglich.

Nach dem Einschalten ist die Steuerung im Zustand ”G122 X0”, wenn nicht in MK_VOREINSTELLUNG ein anderer Modus gewählt wurde.

Beispiel

N33 G122 X1	Verfahrtasten wirken wie Koordinatenverschiebung

3.2.2.50 G125 Nichtmodale Vergleichsoperation, Parameterfeldvergleich

Nichtmodale Vergleichsoperation ohne Beeinflussung anderer, modaler Vergleichsoperationen

Syntax

G125 X Z K I E

Bedeutung der Adressen

X	Operand 1
Z	Operand 2
K	Operator
I	Satznummer des Sprungziels
E	P-Feld-Index zur Ablage des Vergleichsergebnisses muss angegeben werden, falls I nicht programmiert ist

Erläuterung

Die G125 führt einen Vergleich K zwischen den beiden Operanden X und Z durch. Die unter K möglichen Vergleichsoperationen sind

0 : $X = Z$	X gleich Z
1 : $X > Z$	X größer Z
2 : $X < Z$	X kleiner Z
3 : $X \geq Z$	X größer oder gleich Z
4 : $X \leq Z$	X kleiner oder gleich Z
5 : $X \neq Z$	X ungleich Z
6: $(X \wedge 2^Z) = 0$	Bit Z in X gelöscht
7: $(X \wedge 2^Z) \neq 0$	Bit Z in X gesetzt

Trifft die programmierte Bedingung zu (das Vergleichsergebnis ist ”wahr”), wird das Programm an der bezeichneten Stelle (I) fortgesetzt. Andernfalls wird mit dem der G125 nachfolgenden Satz fortgefahren. Im Unterschied zur G130 führt ein ”wahres” Ergebnis nicht zum Löschen modaler Vergleichsoperationen.

Falls E programmiert ist, wird das Vergleichsergebnis (0 oder 1) zusätzlich im angegebenen Parameter gespeichert.

Als Operanden sind sowohl Konstanten, als auch Indizes von Parameterfeldern zulässig.

Beispiel

G125 X=P701 Z100 K1 I20	Wenn der Wert in P701 größer als 100 ist, wird die Programmbearbeitung bei Satz 20 fortgesetzt, sonst mit dem der G125 folgenden Satz.

3.2.2.51 G130 Modale Vergleichsoperation, Parameterfeldvergleich

Modale oder nicht-modale Vergleichsoperation mit Beeinflussung anderer modaler Vergleichsoperationen

Syntax

G130 X Z K I E Y V J

Bedeutung der Adressen

X	1. Operand
Z	2. Operand
K	Vergleichsoperation
I	Zielsatznummer
E	Zielprogrammnummer
Y	Nummer des modalen Vergleichs (0 ... 6) oder falls nicht programmiert, nicht modaler Vergleich.
V	Index einer optionalen kundenspezifischen Vergleichsoperation
J	Bremsen 0: ohne Rampe 1: mit Rampe (Default)

Erläuterung

Die G130 führt einen Vergleich K zwischen den beiden Operanden X und Z durch. Als Operanden sind sowohl Konstanten als auch Indizes von Parameterfeldern zulässig. Die unter K möglichen Vergleichsoperationen sind

0 : $X = Z$	X gleich Z
1 : $X > Z$	X größer Z
2 : $X < Z$	X kleiner Z
3 : $X \geq Z$	X größer oder gleich Z
4 : $X \leq Z$	X kleiner oder gleich Z
5 : $X \neq Z$	X ungleich Z
6: $(X \wedge 2^Z) = 0$	Bit Z in X gelöscht
7: $(X \wedge 2^Z) \neq 0$	Bit Z in X gesetzt

Bei einem ”wahren” Ergebnis werden alle laufenden Aktionen im aktuellen NC-Kanal angehalten. Danach sind zwei Varianten der Programmfortführung möglich:

Programmsprung: Die angehaltenen Aktionen werden abgebrochen und das Programm wird mit Satz I fortgesetzt. In diesem Fall darf der Kennbuchstabe E nicht programmiert sein. Die Zielsatznummer I muss im gleichen Programm zu finden sein, in dem auch der Vergleich programmiert ist. Wenn die Zielsatznummer I ungleich der aktuellen Satznummer N ist, führt ein ”wahres” Ergebnis zum Löschen aller modalen und nicht-modalen Vergleichsoperationen, die auf gleichen bzw. darunter liegenden Programmebenen gemacht wurden.

Unterprogrammaufruf: Die angehaltenen Aktionen werden gerettet, das Programm E wird zur Bearbeitung gebracht. In diesem Fall muss der Kennbuchstabe E programmiert sein; eine Zielsatznummer I darf nicht angegeben werden. Ist das Unterprogramm abgearbeitet, setzt die Steuerung die Bearbeitung an der unterbrochenen Stelle fort. Ein ”wahres” Ergebnis führt nicht zum Austrag von modalen oder nicht-modalen Vergleichsoperationen.

Es liegt im Verantwortungsbereich des Anwenders, alle Achsen, die er im Unterprogramm benutzt, wieder an die Positionen zurück zu bringen, welche er bei Unterprogrammaufruf vorfand.

Sieben modale Vergleiche sind gleichzeitig aktivierbar. Mit Y ist hierbei der Index für die Tabellenzeile anzugeben. Ein fehlendes Y wird als nicht-modaler Vergleich interpretiert, d. h. die Funktion wird nur einmal interpretiert und bleibt nicht im Hintergrund wirksam.

Ein programmierter modaler Vergleich bleibt solange aktiv, bis

- ► ein anderer Vergleich an die gleiche Stelle der Vergleichstabelle programmiert wurde (Y),
- ► der Eintrag mit G131 ausgetragen wird,
- ► dieser oder ein anderer Vergleich (G130, G150, G151) auf der gleichen oder einer höheren Programmebene wahr wird und das Ergebnis des Vergleichs als Satzsprung ausgewertet wird,
- ► Unterprogrammende; bei Rückkehr in eine höhere Programmebene nach Unterprogrammende werden alle Vergleiche deaktiviert, die in diesem Unterprogramm aktiviert wurden,
- ► der Vergleich vollzogen wurde (bei nicht-modalem Vergleich),
- ► Programmende bzw. Programmabbruch.

Hinweis!

Sind in der Vergleichstabelle zu einem bestimmten Zeitpunkt mehrere Bedingungen erfüllt, die zu einem Satzsprung führen würden, so wird nur der modale Vergleich mit dem niedrigsten Tabellenindex zur Ausführung gebracht.

Über die Adresse ”J” kann festgelegt werden, ob die Achsen bei Eintreten des Ereignisses und daraus folgender Unterbrechung einer Verfahrbewegung mit oder ohne Rampe gebremst werden sollen. Die Standardeinstellung (J nicht programmiert) ist ”Bremsen mit Rampe”.

Beispiel

G130 X=P1 Z200 K2 Y1 I30	Die Vergleichsbedingung mit dem Index 1 wird eingetragen. Der Sprung zu Satz 30 wird ausgeführt, wenn der Inhalt von P1 kleiner 200 wird.
N20 G130 X=P1300 K5 Z1 I20	In Satz 20 warten solange P1300 ungleich 1 ist.

3.2.2.52 G131 Modale Vergleichsoperation löschen

Syntax

G131 Y

Bedeutung der Adressen

Y	Nummer des modalen Vergleichs (0 ... 6)

Erläuterung

Mit der Wegbedingung G131 können modale Vergleichsoperationen, die mit G130 aktiviert wurden, wieder aus der Vergleichsoperationstabelle gelöscht werden. Dazu muss unter der Adresse Y die Nummer der Vergleichsoperation programmiert werden.

Falls unter der Y-Adresse eine Nummer größer als 6 programmiert wird oder keine Nummer angegeben wird, erfolgt ein Programmabbruch mit einer Fehlermeldung.

Beispiel

G131 Y1	Die im vorigen Beispiel gesetzte Vergleichsoperation mit dem Index 1 wird wieder gelöscht.

G131 Y1

Die im vorigen Beispiel gesetzte Vergleichsoperation mit dem Index 1 wird wieder gelöscht.

3.2.2.53 G132/G133 Modale Programmverzweigung sperren / erlauben

Syntax

G132
G133

Erläuterung

Mit der G132 können modale Vergleichsoperationen, die mit G130 oder G150 aktiviert wurden vorübergehend gesperrt werden, um kritische Programmstellen vor einer Unterbrechung zu schützen.

Solange die modalen Vergleichsoperationen gesperrt sind, werden weder modale noch nicht-modale Vergleiche mit G130, G150 oder G151 ausgeführt. Modale Vergleiche werden zwar in die Tabelle aufgenommen, werden jedoch erst nach einer G133 ausgeführt. Nicht-modale Vergleiche werden ignoriert.

Bedingte Programmverzweigungen mit G20, G22 oder G125 werden nicht beeinflusst.

Das Sperrung der Vergleichsoperationen wird durch G133, Programmende oder -abbruch wieder aufgehoben.

Beispiel

G132	Keine Unterbrechung jetzt
G0 X100 Y200 S0	Z-Achse in Werkzeugwechselposition fahren
G0 Z0 M10	Werkzeug holen
G0 Z20	und anheben
G0 X0 Y0	in Ausgangsposition fahren
G133	Unterbrechung wieder erlaubt

3.2.2.54 G134 Nicht-modale Wartefunktion, Parameterfeldvergleich

Warten auf das Wahrwerden einer Vergleichsoperation.

Syntax

G134 X Z K

Bedeutung der Adressen

X	1. Operand
Z	2. Operand
K	Vergleichsoperation

Erläuterung

Die G134 führt einen Vergleich K zwischen den beiden Operanden X und Z durch. Als Operanden sind sowohl Konstanten als auch Indizes von Parameterfeldern zulässig.

Im Gegensatz zu G130 erfolgt bei einem positiven Vergleichsergebnis keine Programmverzweigung. Die G134 blockiert lediglich die weitere Programmausführung im aktuellen Kanal bis das Vergleichsergebnis wahr wird.

Die unter K möglichen Vergleichsoperationen sind:

0: X = Z	X gleich Z
1: X > Z	X größer Z
2: X < Z	X kleiner Z
3: X ≥ Z	X größer oder gleich Z
4: X ≤ Z	X kleiner oder gleich Z
5: X ≠ Z	X ungleich Z
6: $(X \wedge 2^Z) = 0$	Bit Z in X gelöscht
7: $(X \wedge 2^Z) \neq 0$	Bit Z in X gesetzt

Die Funktion ist nicht zeitsynchronisiert und daher auch für die schnelle Reaktionen im Bereich von maximal einem Interpolationstakt geeignet. Es werden keine vorher mit G130 oder G150 programmierten modalen Vergleiche ausgetragen.

Beispiel

G134 X=P16 K3 Z50	Wartet darauf, dass P16 ≥ 50 wird.
...	Sätze, die anschließend ausgeführt werden sollen

3.2.2.55 G140/G141/G142 Unrundschleifen ein-/ausschalten

Die Funktionen G140 ... G142 dienen zum Ein- bzw. Ausschalten der Funktion Unrundschleifen.

Syntax

G140
G141 X C D L K E
G142 X C D L K E

Bedeutung der Adressen

X	Gesamtes Aufmaß in mm
C	Anfangsbereich der Zustellung als Position der Winkelachse (nur bei L=0). Wenn nicht programmiert, wird der Anfangsbereich der Zustellung durch die Startposition der C-Achse bestimmt.
D	Schleifscheibendurchmesser. Wenn nicht programmiert, wird der aktuelle Durchmesser aus den Werkzeugdaten übernommen.
L	Kontur 0: ist stetig (Default) 1: ist unstetig
K	Unrundschleifen 0: ohne Korrekturtabelle (Default) 1: mit Korrekturtabelle (siehe G144, G145)
E	Unrundschleifen 0: von Außen (Default) 1: von Innen

Erläuterung

Mit G141 wird die Funktion Unrundschleifen eingeschaltet, und zwar mit negativer Drehrichtung der C-Achse während der Konturerzeugung. Die G142 aktiviert Unrundschleifen mit positiver Drehrichtung.

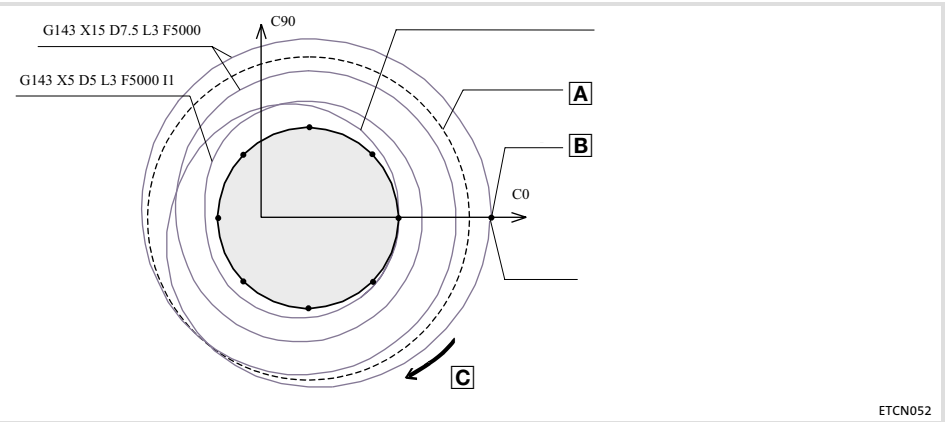
Die G140 schaltet die Funktion Unrundschleifen wieder aus.

Zwischen der G141/G142 und der G140 erfolgt der Aufruf der einzelnen Schleifphasen durch G143 mit der Beschreibung der Fertigkontur als Unterprogramm. Die Drehrichtung der Konturbeschreibung muss mit der durch G141 bzw. G142 vorgegebenen Drehrichtung übereinstimmen.

Beispiel

N10 G31	Bahnbetrieb
N20 G01 X50 C0 F5000	Startposition anfahren
N30 G60 X0	Look-Ahead einschalten
N40 G142 D500 X20 L0 E0	Unrundschleifen für glatte Kontur einschalten
N41 G5 X24.142 Y-14.142 I3 L1	Spline-Interpolation für geschlossene Kurve einschalten
N42 G143 X15 D7.5 F5000 L3 I0	2 Umdrehungen Vorschleifen
N43 G143 X5 D5 F5000 L3 I1	Fertigschleifen mit einer Ausfeuerumdrehung
N44 G143 X-20 D20 F5000 L3	Abfahren
N45 G5	Spline-Interpolation ausschalten
N46 G140	Unrundschleifen ausschalten
N50 G60	Look-Ahead ausschalten
N60 M30	Programmende
%3	
N10 X24.142 Y14.142	Unterprogramm zur Beschreibung der Fertigkontur
N20 X10 Y20	
N30 X-4.142 Y14.142	
N40 X-10 Y0	
N50 X-4.142 Y-14.142	
N60 X10 Y-20	
N70 X24.142 Y-14.142	
N80 X30 Y0	
N90 G99	

Position der X-Achse im kartesischen Koordinatensystem:



A Anfangsaufmaß
B Startpunkt
C Drehrichtung des Werksstücks

Das Werkstück wird erst während der ersten Umdrehung angefahren, da das tatsächliche Aufmaß kleiner als das vorgegebene Aufmaß ist.

Das Abfahren von der Fertigkontur erfolgt auch über G143, indem das Aufmaß und der Zustellbetrag negativ programmiert werden.

3.2.2.56 G143 Parameter von Schleifphasen beim Unrundschleifen

Mit der G143 müssen für jede Schleifphase, z. B. Schruppen, Schlichten, Feinschlichten und Ausfeuern die Parameter programmiert werden.

Syntax

G143 X D C F L J I

Bedeutung der Adressen

X	Aufmaß für diese Phase in mm.
D	Zustellbetrag pro Umdrehung in mm.
C	Bewegungsbereich der C-Achse in dem die Zustellung erfolgen soll. Falls nicht programmiert erfolgt die Zustellung über eine ganze Umdrehung der C-Achse.
F	Bahngeschwindigkeit während dieser Phase in mm/min.
L	Programmnummer des Unterprogramms, in dem die Fertigkontur beschrieben ist.
J	Satznummer im aktuellen Programm, ab dem die Fertigkontur beschrieben ist.
I	Anzahl von zusätzlichen Umdrehungen ohne Zustellung.

Erläuterung

Die G143 wird benötigt, um die Schleifphasen beim Unrundschleifen zu definieren. Mindestens eine Schleifphase ist notwendig.

Bei L wird die Nummer des Programms angegeben, in dem die Fertigkontur beschrieben ist. Alternativ zu L kann auch mit J die Satznummer, ab der die Fertigkontur im aktuellen Programm beschrieben ist, angegeben werden. In jedem Fall muss die Beschreibung der Fertigkontur mit G99 beendet werden.

Die Kontur wird in einer virtuellen X/Y-Ebene beschrieben, wobei der X-Position 0° und der Y-Position 90° der C-Achse entspricht. Dazu können sowohl kartesische Koordinaten (G1, G2, G3) als auch Polarkoordinaten (G101, G102, G103) verwendet werden.

Bei Verwendung der Spline-Interpolation darf die G5 nicht innerhalb der Fertigkontur programmiert werden, sondern muss bereits vor der ersten G143 eingeschaltet und nach der letzten G143 ausgeschaltet werden.

Die gesamte Anzahl der Werkstückumdrehungen einer Schleifphase ergibt sich aus dem programmierten Aufmaß und dem Zustellbetrag pro Umdrehung (X/D) plus der programmierten Anzahl von Umdrehungen ohne Zustellung.

Beispiel

Siehe G140/G141/G142

3.2.2.57 G144/G145 Programmieren einer Korrekturtabelle beim Unrundschleifen

Die Korrekturtabelle dient zur Kompensation beliebiger systematischer Fehler bei der Erzeugung einer Kontur beim Unrundschleifen.

Syntax

G144 D
G145 C X

Bedeutung der Adressen

D	Abstand der Stützpunkte in der Tabelle in Grad
C	Wert der Bezugsachse in Grad
X	Korrekturwert in µm

Erläuterung

Mit der G144 wird die Korrekturtabelle konfiguriert und alle Korrekturwerte auf 0 initialisiert. Die Stützpunkte werden durch die nachfolgenden G145-Befehle definiert. Dabei müssen nicht alle 720 möglichen Stützpunkte programmiert werden. Die Korrekturwerte der nicht programmierten Stützpunkte werden auf 0 initialisiert.

Die Tabelle enthält die Korrekturwerte für die X-Achse in Bezug auf die Modulo-360°-Position der C-Achse. Bis zu 720 Korrekturwerte können in der Tabelle gespeichert werden, d. h. der minimale Abstand zwischen den Stützpunkten beträgt 0,5°.

Die Korrektur erfolgt durch Addition der Korrekturwerte auf die Achssollwerte der X-Achse. Zwischen zwei benachbarten Stützpunkten erfolgt eine lineare Interpolation.

In der Anzeige der NC-Ist-Position erscheinen immer die unkorrigierten Positionswerte.

Die Korrekturtabelle muss vor dem Einschalten des Unrundschleifens definiert werden.

Beispiel

G144 D10	Abstand der Stützpunkte soll 10° betragen
G145 C330 X12	Die Korrektur erfolgt im Bereich von 320° bis 20°, wobei der Korrekturwert bei 320° und bei 20° 0 µm. Bei 350° erfolgt mit 13 µm die größte Korrektur.
G145 C340 X13	
G145 C350 X13	
G145 C360 X13	
G145 C10 X12	

3.2.2.58 G150 Modale Vergleichsoperation, Q-Bit Vergleich

Ein- bzw. Austragung eines modalen Vergleiches auf ein externes Ereignis (Q-Bit)

Syntax

G150 X Y Z E V J L

Bedeutung der Adressen

X	Ziel-Satznummer
Y	Ziel-Programmnummer
Z	Zustand des Vergleiches
E	Index des externen Ereignisses
V	Index einer optionalen kundenspezifischen Vergleichsoperation
J	Bremsen 0: ohne Rampe 1: mit Rampe (Default)
L	Parameterfeldindex, zum Speichern der Achs-Ist-Positionen

Erläuterung

Die G150 führt einen Vergleich des Q-Bit E (0 ... 63) auf den Zustand Z (0 bzw. 1) durch. Dies gilt auch für ”schnelle Eingänge”, die ebenfalls als Q-Bit abgebildet sind.

Bei einem ”wahren” Ergebnis werden alle laufenden Aktionen im aktuellen NC-Kanal angehalten. Danach sind bei der G150 zwei Varianten der Programmfortführung möglich:

Programmsprung:

Die angehaltenen Aktionen werden abgebrochen und das Programm mit dem Satz X fortgesetzt. In diesem Fall darf der Kennbuchstabe Y nicht programmiert sein. Die Zielsatznummer X muss im gleichen Programm zu finden sein, in dem auch der Vergleich programmiert ist.

Wenn die Zielsatznummer X ungleich der aktuellen Satznummer N ist, führt ein ”wahres” Ergebnis zum Austrag aller modalen und nicht-modalen Vergleichsoperationen, die auf gleichem bzw. darunter liegenden Programmebenen gemacht wurden.

Unterprogrammaufruf:

Die angehaltenen Aktionen werden gerettet, das Programm Y wird zur Bearbeitung gebracht. In diesem Fall muss der Kennbuchstabe Y programmiert sein; eine Zielsatznummer X darf nicht angegeben werden. Ist das Unterprogramm abgearbeitet, setzt die Steuerung die Bearbeitung an der unterbrochenen Stelle fort. Ein ”wahres” Ergebnis führt nicht zum Austrag von modalen oder nicht-modalen Vergleichsoperationen.

Es liegt im Verantwortungsbereich des Anwenders, alle Achsen, die er im Unterprogramm benutzt, wieder an die Positionen zurückzubringen, welche er bei Unterprogrammaufruf vorfand.

Die Funktion bleibt modal wirksam. Sie wird deaktiviert, wenn

- ► der Eintrag zurückgesetzt wird (siehe unten)
- ► bei positivem Ergebnis des Vergleichs und daraus resultierendem Satzsprung

- ► bei Unterprogrammende; in diesem Fall werden alle Vergleiche deaktiviert, die in diesem Unterprogramm aktiviert wurden.
- ► Eintreten eines anderen modalen Ereignisses (G130, G150) in einer höheren Programmebene mit daraus resultierendem Satzsprung. Da in diesem Fall ein Rücksprung in die höhere Programmebene erfolgen muss, werden alle modalen Vergleiche der darunter liegenden Programmebenen deaktiviert.
- ► Programmende oder Programmabbruch (Stop oder Fehler)

Es sind 7 modale Vergleiche gleichzeitig aktivierbar.

Hinweis!

Es darf nicht gleichzeitig ein modaler Vergleich und ein nicht-modaler Vergleich auf das gleiche Q-Bit mit dem gleichen Ergebnis programmiert werden. Dies führt u. U. zu nicht definierten Ergebnissen.

Über die Adresse J kann festgelegt werden, ob die Achsen bei Eintreten des Ereignisses und daraus folgender Unterbrechung einer Verfahrbewegung mit oder ohne Rampe gebremst werden sollen. Die Standardeinstellung (J nicht programmiert) ist ”Bremsen mit Rampe”.

Wird die Adresse L programmiert, werden beim Zutreffen der mit E und Z definierten Bedingung, die Ist-Positionen aller konfigurierten Achsen ab dem bei L angegebenen Index im Parameterfeld gespeichert. Diese Funktion kann verwendet werden, um die Position des Ereignisses unabhängig von der konfigurierten Bremsrampe zu ermitteln.

Ein Eintrag wird aus der Tabelle der aktiven Q-Bit-Vergleiche ausgetragen, in dem die G150 mit der Angabe der Q-Bit-Nummer E ohne weitere Parameter programmiert wird.

Beispiel

N100 G150 X200 Z1 E22	Verzweigung zu Satznummer 200, wenn im Q-Feld an der Stelle 22 eine 1 erscheint.
N200 G150 X500 E18 Z0 L1780	Verzweigung zu Satz 500, wenn Q-Bit 18 den Zustand 0 hat, zusätzlich werden die Achs-Ist-Positionen ab P1780 abgelegt.
N300 G150 E22	Aufheben der modalen Überwachung von Q-Bit 22.

3.2.2.59 G151 Nichtmodale Vergleichsoperation, Q-Bit Vergleich

Ausführung eines nicht-modalen Vergleiches auf ein externes Ereignis (Q-Bit).

Syntax

G151 X E Z V

Bedeutung der Adressen

X	Sprungziel Satznummer
Z	Zustand des Vergleiches
E	Index des externen Ereignisses
V	Index einer optionalen kundenspezifischen Vergleichsoperation
L	Parameterfeldindex, zum Speichern der Achs-Ist-Positionen

Erläuterung

Die G151 führt einmalig (nicht modal) einen Vergleich des Q-Bit E (0 ... 63) auf den Zustand Z (0 bzw. 1) durch.

Bei einem ”wahren” Ergebnis wird das Programm mit Satz X fortgesetzt. Die Zielsatznummer X muss im gleichen Programm zu finden sein, in dem auch der Vergleich programmiert ist.

Wenn die Zielsatznummer X ungleich der aktuellen Satznummer N ist, führt ein ”wahres” Ergebnis zum Austrag aller modalen Vergleichsoperationen, die auf gleichen bzw. darunter liegenden Programmebenen gemacht wurden.

Über die Adresse J kann festgelegt werden, ob die Achsen bei Eintreten des Ereignisses und daraus folgendem Abbruch einer Verfahrbewegung mit oder ohne Rampe gebremst werden sollen. Die Standardeinstellung (J nicht programmiert) ist ”Bremsen mit Rampe”.

Wird die Adresse L programmiert, werden beim Zutreffen der mit E und Z definierten Bedingung, die Ist-Positionen aller konfigurierten Achsen ab dem bei L angegebenen Index im Parameterfeld gespeichert.

Diese Funktion kann verwendet werden, um die Position des Ereignisses unabhängig von der konfigurierten Bremsrampe zu ermitteln.

Beispiel

N10 G151 X10 E31 Z1	Das Programm wartet mit der Fortführung der Programmabarbeitung, bis das Q-Bit 31 den Zustand 0 angenommen hat (Sprung auf sich selbst, solange Bedingung (Q[31] == 1) zutrifft).

3.2.2.60 G152 Nicht-modale Wartefunktion, Q-Bit Vergleich

Warten auf das Wahrwerden eines Vergleichs auf ein externes Ereignis.

Syntax

G152 E Z

Bedeutung der Adressen

E	Index des externen Ereignisses
Z	Zustand des Vergleiches

Erläuterung

Die G152 führt einen Vergleich des Q-Bits E (0 ... 63) auf den Zustand Z (0 bzw. 1) durch.

Im Gegensatz zu G151 erfolgt bei einem positiven Vergleichsergebnis kein Satzsprung. Die G152 blockiert lediglich die weitere Programmausführung im aktuellen Kanal bis das Vergleichsergebnis wahr wird.

Die Funktion ist nicht zeitsynchronisiert und daher bei Verwendung von schnellen Eingängen auch für die schnellen Reaktionen im Bereich von maximal einem Interpolationstakt geeignet. Es werden keine vorher mit G130 oder G150 programmierten modalen Vergleiche ausgetragen.

Beispiel

G152 E3 Z0	Wartet auf den Zustand 0 von Q-Bit 3
G152 E3 Z1	Wartet auf den Zustand 1 von Q-Bit 3
...	Sätze, die nach einer positiven Flanke von Q-Bit 3 ausgeführt werden sollen

3.2.2.61 G153 Warten auf die Terminierung eines Kanal 1

Die G153 dient zur Synchronisation des Programmendes im Kanal 1 mit dem Kanal 0.

Syntax

G153 K

Bedeutung der Adressen

K	Kanalnummer des Kanal 1

Erläuterung

Die G153 wird im Kanal 0 aufgerufen, um auf das Programmende im Kanal 1 zu warten und damit die Achsen des Kanal 1 wieder an den Kanal 0 zurückzugeben. Die Rückgabe der Achsen ist nur durch die Terminierung des Kanal 1 möglich.

Ein Kanal kann außer durch Programmabbruch nicht von außen terminiert werden, sondern nur durch eine M30 auf Hauptprogrammebene des entsprechenden Kanals.

Beispiel

G22 L10 K1 U0 V0	Starten des Programms %10 im zweiten Kanal mit Übergabe der U- und V-Achse.
G1 X10 Y5 F100	Die Achsen U und V können hier nicht gefahren werden.
G153 K1	Warten bis das Programm im Kanal 1 beendet ist.
G0 U0 V0	Hier können die Achsen U und V wieder angesprochen werden.
M30	
%10	Programm für den zweiten Kanal
G0 U100 M23	
G0 V50 M25	
G0 U200 V150 M27	
M30	

3.2.2.62 G158/G159 Intermittierender Betrieb ein-/ausschalten

Die Wegbedingung G158 schaltet den intermittierenden Betrieb ein, bei dem die Interpolation der programmierten Kontur hubweise ausgeführt wird.

Syntax

G158 I J K
G159

Bedeutung der Adressen

K	Geschwindigkeitsprofil jedes einzelnen Interpolationshubes 0: ohne Intermittieren 1: sprunghaft (default) 2: linear 3: sinusförmig 4: sinusförmig beschleunigen sinusquadratförmig bremsen 5: sinusquadratförmig
I	Vorschubdauer in Prozent der Schrittlänge (default = 50)
J	Phasenverschiebung in Prozent 0: Vorschub am Schrittende (default) 100: Vorschub am Schrittanfang

Erläuterung

Bei eingeschaltetem intermittierendem Betrieb werden alle Interpolationsarten auf der Kontur hubweise ausgeführt. Der mit jedem Hub zurückzulegende Weg auf der Bahn entspricht der Schrittlänge L bei G1/G2/G3. Durch Kombination von G158 und G33 ist es bei einer Nähmaschine (siehe Technologie Nähen) möglich, den Materialtransport nur in einem begrenzten Winkelbereich einer Nähmotorumdrehung durchzuführen.

Jeder Hub bedingt einen Beschleunigungs- und Bremsvorgang des zu transportierenden Materials. Die Art dieser Beschleunigungskurve wird mit der Adresse K unabhängig von der aktuellen Bahnbeschleunigung festgelegt. Die Mechanik wird bei sinusquadratförmiger Beschleunigung am wenigsten belastet. Allerdings ist dabei die doppelte Geschwindigkeit wie bei sprunghafter Beschleunigung notwendig, um den gleichen Weg in derselben Zeit zurückzulegen.

Mit I wird die Dauer des Materialtransports im Verhältnis zu einer vollen Nähmotorumdrehung in Prozent festgelegt. Der Wert 50 entspricht dabei einer halben Umdrehung. Je kleiner der programmierte Wert ist, desto größer werden die daraus resultierenden Werte für Maximalgeschwindigkeit und Beschleunigung.

Der intermittierende Betrieb ist auf Konturen zwischen M14 und M15/M16/M21 wirksam. Er ist modal wirksam, auch über das Programmende hinaus. Er wird abgeschaltet durch G159.

Beispiel

G158 K2	intermittierenden Betrieb mit linearen Rampen aktivieren
G0 X10 Y300 M14	Startposition anfahren, und Konturbeginn markieren (M14)
G0 A90	Nadel in die Startposition 90° bringen (0° = höchste Position)
G33 A0	Automatische Kopplung zwischen Bahn und Rotationsachse einschalten
G1 E500 L3	Drehzahl und Stichlänge auswählen
X300	Kontur
Y10	
X10	
Y300	
M15	Konturende markieren
G34	Bahnkopplung ausschalten
G0 A0	Nadel hoch
G159	intermittierender Betrieb aus

3.2.2.63 G161 Istpositionsübernahme

Übernahme der aktuellen Istpositionen der Achsen als neue Startposition für die nächste Interpolation.

Syntax

G161 AXES

Bedeutung der Adressen

AXES	Gültigkeitskennung der Achsen, deren Istpositionen übernommen werden sollen.

Erläuterung

Sollte es in einem DIN-Programm einmal notwendig sein, bewusst die Reglerfreigabe einer oder mehrerer Achsen vorübergehend wegzunehmen, kann diese Funktion hilfreich sein. Mit der G161 ist es nach dem Wiedereinschalten der Reglerfreigabe möglich, die eventuell geänderten Achspositionen als neue Startposition für die nächste Verfahrfunktion in den Interpreter zu übernehmen.

Sind keine Achsen programmiert, werden die Istpositionen von allen Achsen des aktuellen Kanals übernommen.

Beispiel

M80	SPS nimmt Regler- und Einlesefreigabe vorübergehend weg
G161	Übernahme aller Achsistpositionen

3.2.2.64 G162 Achsverbund definieren

Schaltet die Zwangskopplung zweier Achsen ein bzw. aus.

Syntax

G162 LAX FAX

Bedeutung der Adressen

LAX	Kennbuchstabe der Leitachse, der programmierte Wert muss 0 sein
FAX	Kennbuchstabe der Folgeachse und Kopplungsfaktor ≠0: Achsverbund mit dem angegebenen Kopplungsfaktor definieren 0: Achsverbund löschen

Erläuterung

Die G162 dient zur Definition eines Achsverbundes. Mit dieser Funktion ist es möglich, eine oder mehrere Folgeachsen softwaremäßig über einen frei definierbaren Faktor an eine Leitachse zu koppeln.

Dabei errechnet sich die korrigierte Sollposition der Folgeachse **P_F'** über die folgende Formel: **PF' = PF + m V PL - b**

P_F = Sollposition der Folgeachse
P_L = Sollposition der Leitachse
m = Kopplungsfaktor bei G162
b = Startoffset beim Einschalten der G162

Der Startoffset wird so berechnet, dass sich beim Einschalten der G162 keine Bewegung der Folgeachse ergibt: **$b = m \cdot P_L$**.

Wenn mehrere Folgeachsen an eine Leitachse gekoppelt werden sollen oder umgekehrt, muss die G162 einmal für jedes zu definierende Achsenpaar programmiert werden. Die daraus resultierende Gesamtsollposition inklusiver aller Transformationen (auch G162) und Koordinatensystemoffsets (S0+Sn+Tn) ist ab P944 für alle Achsen im Parameterfeld abgelegt.

Wenn keine Achse programmiert ist, werden alle mit G162 aktivierten Achskopplungen ausgeschaltet.

Hinweis!

Die Funktion führt keine Geschwindigkeitsbegrenzung der Folgeachse durch und die Istwertdifferenz zwischen Leitachse und Folgeachse wird nicht überwacht. Falls notwendig, muss eine entsprechende Überwachung in der SPS durchgeführt werden (siehe Beispiel).

Wenn die Vorschubfreigaben der Folgeachsen vorhanden sind, können diese Achsen auch bei eingeschalteter Kopplung unabhängig von der Leitachse verfahren werden. Dies kann beim Einrichten der Maschine u. U. hilfreich sein.

Beispiel

Die Achsen A und B sollen nach der Referenzpunktfahrt eins zu eins an die Bewegung der Achse C angekoppelt werden. Die Startpositionen und Koordinatensystemoffsets aller drei Achsen beim Einschalten sollen gleich sein, d. h. b=0 und m=1.

G162	Achsverbund löschen
G74 C0 A0 B0	Referenzpunktfahrt
G0 C0 A0 B0	alle zu koppelnden Achsen auf die gleiche Startposition fahren
G162 C0 A1	A-Achse an C-Achse koppeln
G162 C0 B1	B-Achse an C-Achse koppeln
M174	SPS informieren, dass Achskopplung erfolgt ist

3.2.2.65 G175 Achstausch

Tausch zweier Achsen bezüglich der Eingabe

Syntax

G175 AXES

Bedeutung der Adressen

AXES	Gültigkeitskennung der beiden Achsbuchstaben, die getauscht werden sollen; Zahlenwert ohne Bedeutung

Erläuterung

Mit G175 kann ein Tausch der Achszuordnung programmiert werden. Es können zwei gleichartige Achsen miteinander getauscht werden, d. h. es können nur Linearachsen mit Linearachsen und Rotationsachsen mit Rotationsachsen getauscht werden. Alle Positionsangaben einer Achse werden nach dem Tausch auf die getauschte Achse bezogen.

Ein nochmaliger Tausch von Achsen ist erst möglich, nachdem durch Programmierung von G175 ohne weitere Angaben der Ursprungszustand wieder hergestellt wurde.

Beispiel

G175 U0 Z0	Ab hier werden alle Wegangaben für ”U” für die ”Z”-Achse interpretiert. An der Maschine fährt die Z-Achse die für U programmierten Wege und umgekehrt.

3.2.2.66 G180 Modales Fahren einschalten

Eine oder mehrere Achsen modal mit der programmierten Geschwindigkeit in die angegebene Richtung verfahren

Syntax

G180 AXES F

Bedeutung der Adressen

AXES	Zu fahrende Achsen. Der programmierte Wert gibt die Verfahrrichtung an. > 0: positive Verfahrrichtung, < 0: negative Verfahrrichtung, = 0: nur Geschwindigkeit vorwählen.
F	Modale Verfahrgeschwindigkeit.

Erläuterung

Mit der G180 können Achsen modal in die angegebene Richtung verfahren werden, bis die Verfahrfunktion mit G181 wieder aufgehoben wird. Die weitere Programmausführung wird dabei nicht blockiert, so dass parallel dazu andere Achsen im Programm verfahren werden können.

Werden mehrere Achsen in einer G180 programmiert, wird die Beschleunigungsrampe der einzelnen Achsen so reduziert, dass alle Achsen gleichzeitig ihre programmierte Geschwindigkeit erreichen. Wenn eine Verfahrrichtung angegeben wird und die Achse bereits modal verfährt, wird die Achse ohne anzuhalten auf die neue Geschwindigkeit und Richtung beschleunigt.

Wird eine Vorschubgeschwindigkeit F programmiert, gilt diese für alle im gleichen Satz programmierten Achsen. Sind Linear- und Rotationsachsen beteiligt, gilt sie jedoch nur für die Linearachsen. Das Vorwählen einer Geschwindigkeit für eine oder mehrere Achsen, ohne diese zu starten, erfolgt durch Vorgabe der ”Verfahrrichtung 0”.

Modal fahrende Achsen können mit Geometriefunktionen nicht positioniert werden. Sie werden durch G181, die Stopptaste und durch Fehler der Klasse 3 und 4 angehalten.

Beispiel

N10 G180 X0 F1000	1000 mm/min als modale Geschwindigkeit für X vorwählen
N20 G180 A0 B0 F20	20 U/min als modale Geschwindigkeit für A und B vorwählen
N30 G180 X1 A1 B-1	X,A und B modal verfahren. X in positive Richtung mit 1000 mm/min, A positiv mit 20 U/min und B negativ mit 20 U/min.
N40 G180 Y-1 F2000	Y mit 2000 mm/min modal in negative Richtung verfahren
...	
N80 G181 X0 Y0	X und Y anhalten, A und B fahren weiter.
N90 G181	alle mit G180 gestarteten Achsen anhalten

3.2.2.67 G181 Modales Fahren ausschalten

Eine, mehrere oder alle mit G180 gestarteten Achsen anhalten

Syntax

G181 AXES

Bedeutung der Adressen

AXES	Anzuhaltende Achsen. Der programmierte Wert ist ohne Bedeutung.

Erläuterung

Alle programmierten Achsen werden angehalten, wenn diese mit G180 gestartet wurden. Um alle mit G180 gestarteten Achsen anzuhalten, genügt es diese Funktion ohne Achsen zu programmieren.

Beispiel

siehe G180

3.2.2.68 G187 Handrad

Ein- und Ausschalten der Handradfunktion für eine oder mehrere Achsen

Syntax

G187 AXES

Bedeutung der Adressen

AXES	Bewertungsfaktor des Handrades für die zugeordnete Achse

Erläuterung

Der unter AXES programmierte Wert wird als Bewertung des Handradweges für die ausgewählte Achse interpretiert. Die Handradfunktion arbeitet überlagert zu allen anderen Verfahrbewegungen der Achsen.

Die Auflösung des Handrads wird mit MK_IMPULSE und MK_WEG konfiguriert. Das Produkt aus dem Bewertungsfaktor und dem Wert aus MK_HANDRADFAKTOR wirkt als Multiplikator auf den Weg des Handrades.

Standardmäßig ist kein Handrad aktiv.

Die Zuordnung der Achsen zu den konfigurierten Handrädern wird mit MK_HANDRADZUORDNUNG festgelegt. Dies ist notwendig, falls mehr als ein Handrad an der Maschine vorhanden ist.

Die Funktion wird standardmäßig intern über die Verfahrtastencodes 105 ... 109 erzeugt.

Beispiel

N30 G187 X0.2	Die Impulse des Handrades werden mit dem Faktor 0,2 bewertet der Position der X-Achse zugeschlagen.
N40 G187 X0	Die Zuordnung des Handrades zur X-Achse aufheben.

3.2.2.69 G193 Absolutes Nullpunktsetzen

Der momentane Standpunkt der programmierten Achsen wird im aktuellen Koordinatensystem auf den angegebenen Positionswert gesetzt.

Syntax

G193 AXES

Bedeutung der Adressen

AXES	Achsen, deren Koordinatennullpunkt gesetzt werden soll.

Erläuterung

Mit der G193 kann der aktuelle Standpunkt der Achsen im eingeschalteten Koordinatensystem auf die bei den Achsen programmierten Werte gesetzt werden. Dies wird hauptsächlich dazu verwendet, um den Koordinatennullpunkt auf die aktuelle Position der Achsen zu eichen. Der momentane Standpunkt der programmierten Achsen wird dabei zum Koordinatennullpunkt gemacht.

Durch Angabe einer Zahl ungleich 0 kann eine gleichzeitige Nullpunktverschiebung vorgenommen werden, d. h. die Achse steht dann auf dem programmierten Punkt. Eventuell gespeicherte Nullpunkte für die angegebenen Achsen werden durch G193 überschrieben.

Beispiel

G193 X0 Z0	Der momentane Standort von X und Z wird zum neuen Koordinatenursprung erklärt.
G193 X10	Der Nullpunkt wird so gesetzt, dass die X-Achse nach der Verschiebung auf dem Wert ”10” steht.

3.2.2.70 G194 Programmieren eines zusätzlichen Grundoffsets

Die G194 dient zum Programmieren eines Grundoffsets, der zusätzlich zu dem Grundoffset in den Maschinenkonstanten wirkt. Die Verschiebung des Grundoffsets erfolgt ohne Berücksichtigung des momentanen Koordinatensystems.

Syntax

G194 AXES

Bedeutung der Adressen

AXES	Achsen, deren Grundoffset gesetzt werden soll.

Erläuterung

Der bei G194 programmierte Grundoffset wird zum Wert aus MK_GRUNDOFFSET addiert und an die Achsen übermittelt. Im Gegensatz zu einer Koordinatenverschiebung in einem Koordinatensystem darf die G194 auch in einem Interrupt-Programm verwendet werden.

Die Offsetverschiebung erfolgt direkt im Achsrechner unter Umgehung von Interpreter und Handler. Der mit G194 programmierte Grundoffset wird nicht gespeichert und geht beim Ausschalten der Steuerung verloren. Die programmierten Werte werden ab P928 im Parameterfeld abgelegt.

Beispiel

G194 X150	Der Grundoffset der X-Achse wird auf MK_GRUNDOFFSET+150 gesetzt.
G194 X0	Der Grundoffset der X-Achse wird auf den Wert von MK_GRUNDOFFSET rückgesetzt.

3.2.2.71 G195 Absolute Koordinatenverschiebung aller S-Koordinatensysteme

Mit der G195 können gleichzeitig alle S-Koordinatensysteme um den programmierten Betrag in Bezug auf die durch G193/G92 festgelegten Nullpunkte verschoben werden. Die Verschiebung wirkt additiv zu den aktuellen Nullpunkten aller S-Koordinatensysteme außer S0.

Syntax

G195 AXES

Bedeutung der Adressen

AXES	Achsen, deren Nullpunkt verschoben werden soll.

Erläuterung

Mit der G195 wird der Offset von S0 programmiert. Da alle anderen Koordinatensysteme relativ zu S0 sind, bewirkt dies effektiv eine Verschiebung aller anderen Koordinatensysteme.

Mit der Maschinenkonstante MK_S0T0_VERSATZ_ERLAUBT kann ausgewählt werden, ob diese Verschiebung auch in S0 wirksam wird. Bei der Defaulteinstellung von 0 wird eine Verschiebung von S0 nicht in S0 wirksam, sondern nur in allen anderen Koordinatensystemen. In diesem Fall wird der Gesamt-Offset (P192ff) der Achsen wie folgt gebildet:

In S0: Offset = Tm

In Sn: Offset = S0+Sn+Tm mit n > 0

Falls MK_S0T0_VERSATZ_ERLAUBT auf 1 gesetzt ist, unterscheidet sich die Bildung des Gesamt-Offsets:

In S0: Offset = S0+Tm

Die Verschiebung mit der G195 ist eine absolute Verschiebung, d. h. der bei den Achsen programmierte Wert ersetzt den alten Offset. Die Verschiebung wird remanent im CMOS-RAM gespeichert, wenn die Maschinenkonstante MK_NULLPUNKTE_SPEICHERN auf 1 gesetzt ist. Sie ist dann auch noch nach dem Aus-/Einschalten der Steuerung wirksam.

Beispiel

G195 X10	Alle Koordinatensystemen außer S0 werden in X um 10 mm verschoben (MK_S0T0_VERSATZ_ERLAUBT = 0).
G195 X0	Verschiebung der X-Achse wird wieder aufgehoben.

3.2.2.72 G200 Ein-/Ausschalten des Geometriefilters

Mit der G200 kann ein Filter aktiviert werden, mit dem das Rauschen der programmierten Kontur unterdrückt werden kann. Verrauschte Konturen entstehen meist, wenn die Punktdichte in der Größenordnung der Auflösung der vorgegebenen Konturstützpunkte liegt.

Syntax

G200 AXES

Bedeutung der Adressen

AXES	Achsen, die für die Filterfunktion berücksichtigt werden sollen mit Angabe der Filterkonstante je Achse.

Erläuterung

Das Geometriefilter kann mehrere G1-Sätze durch eine einzige G1 ersetzen. Dabei werden bis zu 20 Stützpunkte gespeichert und mit jedem neuen Stützpunkt eine Gerade zwischen dem ersten und letzten Punkt berechnet. Dann werden die Abweichungen zwischen der berechneten Gerade und allen gespeicherten Stützpunkten bestimmt und mit den bei G200 programmierten maximal zulässigen Toleranzen verglichen.

Ist die Abweichung in allen Punkten innerhalb der Toleranz, wird der neue Punkt abgespeichert und der zugehörige Satz aus dem Eingabestrom entfernt. Andernfalls wird der aktuelle Satz abgespeichert und das bisherige Optimum der gespeicherten Stützpunkte als neuer Satz ausgegeben.

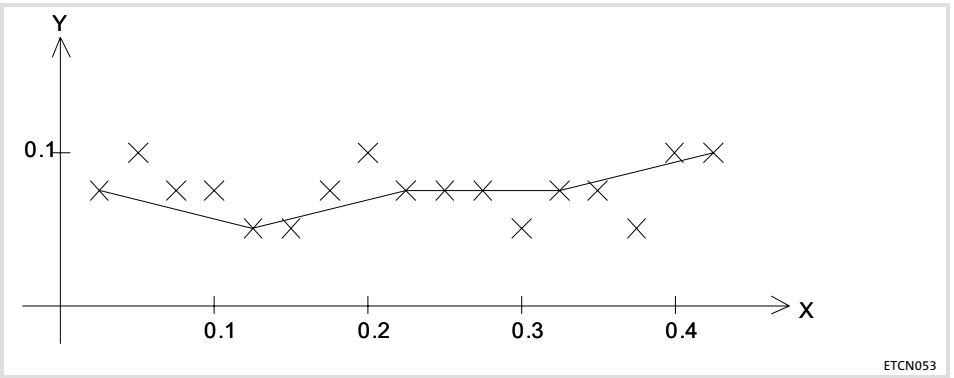
Es können nur G1-Sätze reduziert werden, in denen nichts außer den bei G200 angegebenen Achsen programmiert ist. Sätze mit ”F”, ”M”, ”S”, ”T” oder ”Q” können generell nicht reduziert werden. Die bei G200 programmierten Toleranzen sind in Eingabeeinheiten anzugeben. Die Toleranzen der einzelnen Achsen dürfen unterschiedlich groß sein und werden achsspezifisch berücksichtigt.

Um die Effizienz des Geometriefilters beurteilen zu können wird in P567 das Verhältnis von reduzierbaren zu reduzierten Sätzen in Prozent angezeigt. Die maximal erreichbare Effizienz liegt bei etwa 94 %. Der zeitliche Overhead des Geometriefilters wird ab einer Effizienz von ca. 15 % durch die geringere Anzahl auszuführender Sätze vollständig kompensiert.

Der Geometriefilter wird durch Programmierung von G200 ohne Achsen ausgeschaltet.

Beispiel

G200 X0.1 Y0.05	Ausgabe der Kontur unterdrücken bis die Summe der Konturstücke in X 0,1 mm oder in Y 0,05 mm überschreitet.



3.2.2.73 G201 Ändern der Beschleunigungs- und Bremsrampen

Mit der G201 können die Beschleunigungs- und Bremsrampen im Bahn- und im Streckenbetrieb programmiert werden.

Syntax

G201 AXES J I

Bedeutung der Adressen

AXES	Achsen, deren Rampen im Streckenbetrieb geändert werden soll.
J	Zum Ändern der Rampen im Bahnbetrieb.
I	Welche Rampen sollen geändert werden 0: Beschleunigungs- und Bremsrampe (Default) 1: nur Beschleunigungsrampe 2: nur Bremsrampe

Erläuterung

Zum Ändern der Bahnrampen wird bei J die Höhe der Beschleunigungs- und Bremsrampe in m/s2 programmiert. Der eingegebene Wert wird auf den Wert von MK_BAHNBEACHL bzw. MK_BAHNBREMS in den Maschinenkonstanten begrenzt.

Die Beschleunigungs- und Bremsrampen der einzelnen Achsen im Streckenbetrieb werden bei den entsprechenden Achsbuchstaben bei Linearachsen in m/s2 und bei Rotationsachsen in 1/s2 programmiert. Die eingegebenen Werte werden auf die Werte von MK_BESCHL bzw. MK_BREMS in den Maschinenkonstanten begrenzt.

Mit dem Buchstaben I kann ausgewählt werden ob nur die Beschleunigungsrampen, nur die Bremsrampen oder beide geändert werden sollen. Wenn kein I programmiert ist, werden beide Rampen auf den programmierten Wert geändert.

Wird die G201 ohne Parameter programmiert, werden die Rampen auf der Bahn und aller konfigurierten Achsen auf die in den Maschinenkonstanten eingestellten Werte zurückgesetzt.

Beispiel

G201 J1.5 I1	Beschleunigungsrampe auf der Bahn auf 1,5 m/s^2 ändern.
G201 J2.5 I2	Bremsrampe auf der Bahn auf 2,5 m/s^2 ändern.
G201	Alle Rampen auf die in den MK's konfigurierten Werte zurücksetzen.

3.2.2.74 G209 Setzen des Geometriezählers

Die G209 dient zum Initialisieren des Geometriezählers auf einem benutzerspezifischen Wert.

Syntax

G209 E

Bedeutung der Adressen

E	Neuer Wert des Geometriezählers (default = 0).

Erläuterung

Der Geometriezähler wird im DIN-Display des ETC-MMI zur Synchronisation der Fortschrittsanzeige mit der Programmbearbeitung in der NC-Steuerung benötigt. Er wird von der Steuerung beim Programmstart auf 0 gesetzt und mit jedem neuen Geometriesatz um 1 hochgezählt.

In manchen Fällen mag es sinnvoll sein, diesen Geometriezähler vom NC-Programm aus zu manipulieren. Mit der G209 kann der Geometriezähler des aktuellen NC-Kanals auf einen definierten Wert gesetzt werden.

Der aktuelle Wert des Geometriezählers wird im Parameterfeld eingeblendet, und zwar in P565 für Kanal 0 und in P597 für Kanal 1.

Beispiel

P2000=P565	aktuellen Wert des Geometriezähler merken
...	Verfahrsätze, die nicht mitgezählt werden sollen
G209 E=P2000	alten Wert restaurieren

3.2.2.75 G211 Transformation für zweiachsige Knickarmroboterkinematik

Die G211 dient zum Ein-/Ausschalten einer Transformation für eine einfache zweiachsige Knickarm-roboterkinematik (A,B) mit Unterstützung für eine optionale dritte Achse (C) für die Werkzeugorientierung.

Syntax

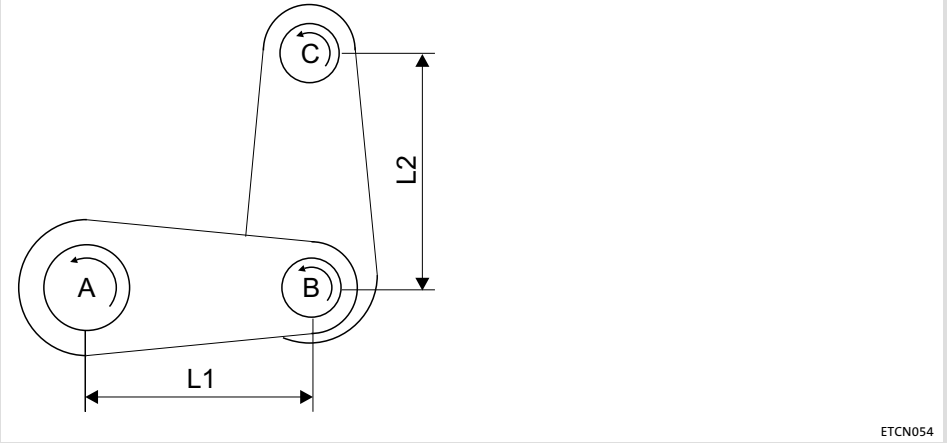
G211 E I J

Bedeutung der Adressen

E	Zum Einschalten der optionalen dritten Achse 0: ohne dritte Achse 1: mit dritter Achse für die Werkzeugorientierung
I	Hebellänge L1, Defaultwert aus P760 (MK_TECHNOLOGIEDATEN1[0])
J	Hebellänge L2, Defaultwert aus P761 (MK_TECHNOLOGIEDATEN1[1])

Erläuterung

Die Funktion transformiert von der X-Y-Ebene im kartesischen Koordinatensystem in ein A-B-Maschinenkoordinatensystem. Dabei wird die Werkzeugorientierung der C-Achse, wenn gewünscht (E1), ebenfalls mit transformiert. Das nachfolgende Bild zeigt die Maschinenkinematik in der 0°-Stellung von A- und B-Achse.



Die Hebellängen L1 und L2 dürfen auch negativ sein.

Programmiert wird bei eingeschalteter Transformation die X/Y-Koordinate des Drehpunktes der C-Achse. In der Ausgangsstellung A0, B0 entspricht dies der Koordinate X0, Y0 im kartesischen Koordinatensystem.

Beim Einschalten der Transformation werden die zwei virtuellen Achsen X und Y angelegt, welche dann zur Programmierung im kartesischen Koordinatensystem dienen. Maximalgeschwindigkeit und Rampen dieser Achsen werden aus den MK's MK_BAHNVMAX, MK_BAHNBESCHL und MK_BAHNBREMS übernommen.

Bei eingeschalteter Transformation können die physikalischen Achsen A und B nur noch indirekt über X,Y bewegt werden. Dabei erfolgt keine Geschwindigkeits- und Beschleunigungsbegrenzung, der an der Transformation beteiligten physikalischen Achsen. Dies ist besonders beim Durchfahren des kritischen Bereichs der Maschine zu beachten. Das Durchfahren von X/Y-Koordinaten in der Nähe des Drehpunktes der A-Achse wird zu sehr hohen Verfahrgeschwindigkeiten der physikalischen Achsen führen!

Für die virtuellen Achsen werden zwei freie Applikationsachsnummern benötigt. Empfehlenswert ist es, die ersten beiden Achsnummern in den MK's für die virtuellen Achsen freizuhalten. In diesem Fall bekommt X die Achsnummer 0 und Y die Achsnummer 1 zugewiesen.

Ausgeschaltet wird die Transformation durch Programmierung von G211 ohne E oder automatisch am Programmende oder bei Programmabbruch.

Beispiel

G211 E1	Transformation mit Berücksichtigung der C-Achse einschalten.
...	Kontur mit X,Y und C-Koordinaten.
G211	Transformation ausschalten.
...	Ab hier sind wieder A,B-Koordinaten erlaubt.

3.2.2.76 G226 Hardwareendschalter umkonfigurieren

Mit der G226 kann die Reaktion der Steuerung beim Fahren auf die richtungsabhängigen Hardwareendschalter einzelner Achsen geändert werden.

Syntax

G226 AXES

Bedeutung der Adressen

AXES	Kennung der Achsen, deren Endschalterfunktion geändert werden soll. 0: Endschalter deaktivieren 1: Normale Endschalterfunktion aktivieren 2: Die Endschalter wirken verfahrbereichsbegrenzend für die entsprechende Achse, führen jedoch nicht zur Fehlermeldung oder zum Programmabbruch.

Erläuterung

Die Kennung, die bei den Achsen programmiert wird, bestimmt, wie die Steuerung auf die Endschalter der entsprechenden Achse reagieren soll. Bei deaktivierten Endschaltern reagiert die Steuerung gar nicht auf die Endschalter der Achse. Bei der normalen Endschalterfunktion reagiert die Steuerung, in dem die Verfahrbewegung ohne Verzögerung angehalten und ein eventuell laufendes NC-Programm abgebrochen wird.

Wenn die Kennung 2 programmiert ist, reagiert die Steuerung auf die Endschalter, in dem sie die Verfahrbewegung der betroffenen Achse in diese Richtung ohne Verzögerung anhält und die Position der Achse einfriert. Die Position bleibt solange eingefroren, bis die Achse die eingefrorene Position in Gegenrichtung wieder verlassen soll. Dabei wird weder eine Fehlermeldung erzeugt noch das Programm abgebrochen. Diese Funktion bewirkt ein Abschneiden von Konturbereichen beim Fahren auf die Endschalter.

Um die Defaulteinstellung aus den Maschinenkonstanten wieder zu aktivieren, kann die G226 ohne Achsen programmiert werden.

Beispiel

N10 G226 X0	Endschalter der X-Achse abschalten.
N90 G226	Voreingestellte Endschalterfunktion für alle Achsen aktivieren.

3.2.2.77 G233 2D/3D-Achsenkorrektur

Bei der 2D/3D-Achsenkorrektur wird eine Achse in Abhängigkeit von ein oder zwei Achsen über eine von maximal 3 Stützpunkttabellen korrigiert.

Syntax

G233 AXE

Bedeutung der Adressen

AXE	Kennung der Achse, die korrigiert werden soll. 0: Korrekturtabellen für die angegebene Achse ausschalten 1: Korrekturtabelle 1 einschalten 2: Korrekturtabelle 2 einschalten 3: Korrekturtabelle 3 einschalten

Erläuterung

Mit dieser Funktion wird die Position einer Achse in Abhängigkeit von ein bis zwei Basisachsen über eine zuvor geladene Tabelle mit äquidistanten Stützpunkten korrigiert. Dabei darf die zu korrigierende Achse eine der beiden Basis-Achsen sein. Die Korrektur erfolgt um den in der Tabelle angegebenen Offset. Zwischen den Stützpunkten wird linear interpoliert.

Der Werkzeug-Offset des aktuellen Werkzeugs Tn wird beim Einschalten der Funktion mit G233 verrechnet und darf daher nicht schon in der geladenen Stützpunkttabelle berücksichtigt werden. Die Koordinaten der Stützpunkttabelle müssen auf das Referenzkoordinatensystem inklusive Grundoffset, d. h. auf S0 ohne Verschiebung, bezogen sein.

Bis zu 3 solcher Korrekturtabellen können geladen werden und sich gleichzeitig überlagern. Dabei kann auch mehrfach die gleiche Zielachse beteiligt sein. Wenn mehrere Korrekturtabellen aktiviert werden, ist die Ausführungsreihenfolge der Korrekturen wichtig:

Tabelle 1: Korrektur von Y in Abhängigkeit von X

Tabelle 2: Korrektur von Z in Abhängigkeit von X, Y

Erst 1 dann 2: In die Korrektur von Z fließt die Korrektur von Y mit ein und die Korrekturen überlagern sich.

Erst 2 dann 1: Die Korrekturen sind unabhängig und überlagern sich nicht.

Die Reihenfolge der auszuführenden Korrekturen entspricht der Programmierungsreihenfolge der G233-Funktionen.

Durch Programmierung von G233 ohne Achsbuchstaben und bei Programmende oder Programmabbruch, werden die Korrekturtabellen für alle Achsen ausgeschaltet.

Beispiel

G233 Z2	Korrekturtabelle 2 einschalten, Zielachse in der Tabelle muss Z sein
G233 Y1	Korrekturtabelle 1 einschalten, Zielachse muss Y sein
G233 Y3	Korrekturtabelle 3 einschalten, Zielachse muss Y sein
	Die Reihenfolge ist damit festgelegt: 2,1,3
G233 Z0	Korrektur von Z ausschalten
	Die verbleibende Reihenfolge ist 1,3
G233 Y0	Alle Korrekturtabellen von Y ausschalten

Aufbau der Korrekturdatei

Die Korrekturtabellen sind als Binärdateien aufgebaut und müssen entweder über ein angeschlossenes ETC-MMI oder mit dem Hyper-Terminal unter dem Namen ”ACHS3D??.KOR” in die Steuerung übertragen werden. Die ”?” müssen durch beliebige Zahlen oder Buchstaben ersetzt werden. Die Datei besteht aus einem 128 Byte langen Header, einem Definitionsblock und einem Block mit den Stützpunktwerten für die zu korrigierende Achse.

Header:

Der Header ist 128 Bytes lang und enthält die nullterminierte Liste der Achsbuchstaben aller beteiligten Achsen, sowie im letzten Byte die binäre Nummer der Korrekturtabelle. Dabei definiert der erste Buchstabe die Achse, die in Abhängigkeit der anderen Achsen korrigiert werden soll.

0	1	2	3	4	127	128
	Z	X	Y	0	...	1-3

Diese wird im Folgenden als Korrekturachse und die anderen Achsen als Basisachsen bezeichnet.

Definitionsblock:

Der Definitionsblock beschreibt den Gültigkeitsbereich und die Größe der nachfolgenden Stützpunkttabelle. Er enthält je Basisachse einen FLOAT für Minimal- und Maximalwert in Eingabeeinheiten und einen ULONG für die Anzahl der Stützpunkte zwischen Minimal- und Maximalwert. Die Gesamtzahl der Stützpunkte einspricht dem Produkt der Anzahl der Stützpunkte aller Basisachsen.

0	4	8	12	16	20	24
	X_{min}	X_{max}	X_{anz}	Y_{min}	Y_{max}	Y_{anz}

Länge = Anzahl Bsisachsen x 12

Stützpunkte:

Die Korrekturwerte werden in Eingabeeinheiten in Tabellenform mit je einem FLOAT pro Stützpunkt definiert. Dabei entspricht der erste Stützpunkt der Koordinate, die den Minimalwerten der Basisachsen gehört und der letzte Stützpunkt der Koordinate, die den Maximalwerten der Basisachsen gehört.

0	4	8	12		4k
	Z_{11}	Z_{12}	Z_{13}	...	Z_{1k}

wobei k = Y_{anz}

0	4	8	12		4k
	Z_{21}	Z_{22}	Z_{23}	...	Z_{2k}

0	4	8	12		4k
	Z_{31}	Z_{32}	Z_{33}	...	Z_{3k}

0	4	8	12		4k
	Z_{i1}	Z_{i2}	Z_{i3}	...	Z_{ik}

wobei i = X_{anz}

Definition der verwendeten Datentypen

BYTE	vorzeichenloser 8-Bit-Integer
ULONG	vorzeichenloser 32-Bit-Integer, 4 Byte lang, mit dem niederwertigsten Byte zuerst
FLOAT	IEEE-Single-Precision Floating-Point, 4 Byte lang, mit dem niederwertigsten Byte der Mantisse zuerst.

3.2.2.78 G234 Richtungsabhängige Höhenkorrektur

Bei der richtungsabhängigen Höhenkorrektur wird die Position der angegebenen Achse in Abhängigkeit von der Fahrtrichtung in der aktuellen Ebene über eine Stützpunkttabelle korrigiert.

Syntax

G234 AXE K

Bedeutung der Adressen

AXE	Kennung der Achse, die korrigiert werden soll. 0: Korrektur ausschalten 1: Korrektur einschalten
K	Parameterfeldindex, ab dem die Korrekturtabelle im P-Feld abgelegt ist.

Erläuterung

Mit dieser Funktion kann zum Beispiel eine Z-Achse in Abhängigkeit von der aktuellen Fahrtrichtung in der XY-Ebene um einen Offset korrigiert werden. Die Korrektur ist dabei auf einen Maximalwert von +/-1 Eingabeeinheit (z.B. mm oder Grad) begrenzt.

Bei K wird der Parameterfeldindex programmiert, ab dem die Korrekturtabelle im P-Feld abgelegt ist. Der erste Wert der Tabelle muss die Anzahl der nachfolgenden Tabellenwerte enthalten. Jeder Tabelleneintrag enthält einen Korrektur-Offset für einen definierten Konturwinkel in der aktuell (Zeitpunkt der G234) ausgewählten Ebene. Die Tabellenwerte müssen einen Winkelbereich von 360° abdecken, der erste Tabellenwert gilt für einen Konturwinkel von 0°. Die Tabelle darf maximal 36 Einträge haben (10° pro Korrekturwert). Zwischen den Einträgen wird linear interpoliert.

Ausgeschaltet wird die Korrektur bei Programmende oder Programmabbruch und durch Programmierung von G234 ohne Achsbuchstaben.

Beispiel

P1200=4	Anzahl der Stützpunkte
P1201=0.0	Offset für 0°
P1202=0.1	Offset für 90°
P1203=0.2	Offset für 180°
P1204=0.3	Offset für 270°
G234 Z1 K1200	Korrektur der Z-Achse einschalten
G234	Korrektur ausschalten

3.2.2.79 G251 Sprungantwort einer Achse aufnehmen

Die G251 dient zur Einstellung der Antriebs- und Lageregelungsparameter einer Achse durch Aufnahme einer Sprungantwort und kann nur von einem speziellen Achseinstellungswerkzeug wie dem AXSCOPE benutzt werden. Für den DIN-Programmierer ist diese Funktion im Allgemeinen nicht von Bedeutung.

Syntax

G251 AXES I J K L

Bedeutung der Adressen

AXES	Geschwindigkeitshub der einzustellenden Achse in [mm/min] bzw. [1/min]
I	Sprungdauer in Sekunden
J	Beschleunigungsdauer in Sekunden
K	Anzahl der Sprünge
L	Kennung 0: Aufzeichnung der Master-Achse (Default) 1: Aufzeichnung der Slave-Achse 2: Aufzeichnung von Master- und Slave-Achse

Erläuterung

Die Funktion oszilliert die angegebene Achse K-mal mit der angegebenen Geschwindigkeit, der Beschleunigungsdauer J und der Sprungdauer I hin und her. Am Ende der Bewegung steht die Achse wieder an ihrem Ausgangspunkt. Der Verfahrweg entspricht I * Geschwindigkeit. Die Pause zwischen den Sprüngen beträgt I - J. Wenn K nicht angegeben ist, oszilliert die Achse 1-mal hin und her.

Während der Ausgabe der Sprungfunktion auf die Achse werden Sollgeschwindigkeit, Ist-Geschwindigkeit, Schleppabstand und Stellgröße als 16-Bit-Integer-Datentyp aufgezeichnet und asynchron an den PC gesendet. Dieser kann die aufgezeichneten Informationen grafisch darstellen oder anderweitig auswerten.

Falls die Kennung L2 bei einer Synchronachse angegeben wird, zeichnet die Steuerung die Ist-Geschwindigkeit der Master-Achse, der Slave-Achse, den Synchronabstand zwischen beiden Achsen und die Sollgeschwindigkeit auf.

Beispiel

G251 X1000 I0.2 J0.1	Ausgeben eines Sollwertsprungs mit einem Geschwindigkeitshub von 1 m/min, einer Sprungdauer von 200 ms und einer Beschleunigungsdauer 100 ms.

3.2.2.80 G252 Eingabe eines Wertes über Anzeigegerät (siehe MMI)

Der in der G-Funktion angegebenen P-Feld Parameter wird durch eine interaktive Abfrage an den Bediener neu bestimmt.

Syntax

G252 A B C E X Y Z F

Bedeutung der Adressen

A	Erlaubter Minimalwert des Eingabeparameters
B	Erlaubter Maximalwert des Eingabeparameters
C	Index des einzugebenden Parameters im Parameterfeld der Steuerung
E	Satznummer, die bei Eingabeabbruch angesprungen werden soll
F	Textnummer oder in Verbindung mit ".." eine Formatanweisung
X	Spalte der Anzeige
Y	Zeile der Anzeige
Z	Zeichengröße für die Anzeige

Erläuterung

Die G252 ist eine Funktion zur interaktiven Eingabe eines Wertes über den Bedienrechner. Als Vorgabevorschlag für die Eingabe wird der aktuelle Wert des angesprochenen P-Feldparameters zur Anzeige gebracht.

Für den Wert der Eingabe ist unter A eine Unter- und unter B eine Obergrenze vorgebbar. Fehlen eine oder beide Eingabegrenzen, so werden die entsprechenden Minimal- bzw. Maximalwerte eingesetzt.

Die Adresse C hält den Index des einzugebenden Parameters, in dem der Eingabewert abgelegt werden soll.

Unter der Adresse E kann eine Satznummer programmiert werden, bei der das Programm fortfahren soll, falls die Eingabe mit dem Tastendruck ESC abgebrochen wurde. Bei Abschluss der Eingabe mit ENTER wird das Programm mit dem auf G252 folgenden Satz fortgesetzt.

Mit X ist eine andere Spalten- und mit Y eine andere Zeilenposition für die Textausgabe wählbar. Zusätzlich ist unter Z eine alternative Zeichengröße vorgebbar. Wenn kein X oder Y angegeben ist, erscheint der Text an der im Bedienrechner festgelegten Stelle in festgelegter Größe (applikations-abhängig; Standard: Statuszeile über Softkeys in normaler Größe).

Bei der Eingabe wird der unter F="..." angegebene Text ausgegeben und der gewünschte Parameter im definierten Format entgegengenommen. Das Eingabeformat und die Anzeigeposition des Parameters ist an der gewünschten Stelle im Textstring mit einer Formatangabe gemäß der nachfolgenden Regeln zu kennzeichnen:

%[Flags][Breite][Genauigkeit]Typ

Flags	
<nichts>	rechtsbündig, führende Leerzeichen/Nullen
-	linksbündig, nachfolgende Nullen
+	Vorzeichen immer ausgeben (auch +)
' '	nur neg. Vorzeichen ausgeben
#	bei le, lE, lf : Dezimalpunkt immer ausgeben
bei lg, lG:	Dezimalpunkt immer ausgeben aber ohne folgende Nullen

Breite	
0n	min n-Stellen, Nullen vorangestellt
n	min n-Stellen, " " vorangestellt

Genauigkeit	
<nichts>	6 Stellen bei le,lE,lf
.0	le,lE,lf kein Dezimalpunkt ausgeben
.n	max. n-Nachkommastellen

Typ	
lf	Fließkomma [-]dddd.ddd
le	Fließkomma [-]d.ddd e[±]ddd
lg	wie le oder lf; je nach Genauigkeit automatische Umschaltung der Darstellung
lE	wie le, mit "E" vor dem Exponenten
lG	wie lg, mit "E" vor dem Exponenten
%	das Zeichen % soll angezeigt werden

Beispiel

N10 G252 F="Bitte Wert eingeben: %10.3lf [mm]" A0 C1024 P1024=500

Der Satz N10 holt über die Anzeige einen neuen Wert für das Parameterfeld 1024 (C). Der Wert darf nicht kleiner als 0 (A) eingegeben werden. In der Anzeige erscheint an intern definierter Stelle (kein X und kein Y) der Text

Bitte Wert eingeben: 500 [mm]

Der momentane Wert von P1204 wird als Eingabe vorgeschlagen. Der Cursor (_) steht auf der äußersten rechten Stelle des Eingabefeldes, das eine Breite von 10 Zeichen hat.

Die Eingabe wird abgeschlossen, wenn die Taste ESC gedrückt wurde oder die Taste ENTER betätigt wurde und der aktuelle Eingabewert innerhalb der vorgegebenen Bereichsgrenzen liegt. Eine mit ESC beendete Eingabe verändert den ursprünglichen Wert des Parameterfeldes nicht. Das Programm fährt in jedem Fall (kein E programmiert) mit dem Satz nach Satz N10 fort.

3.2.2.81 G253 Ausgabe eines Kommentars, optional mit Programmabbruch

Der in der G-Funktion angegebene Text wird am angeschlossenen Bedienrechner anzeigt. Bei Angabe einer Fehlernummer kann das laufende Programm zusätzlich abgebrochen werden.

Syntax

G253 A B C E X Y Z F ( )

Bedeutung der Adressen

A	1. Anzeigeparameter
B	2. Anzeigeparameter
C	3. Anzeigeparameter
E	auf dem Bedienrechner anzuzeigende Fehlernummer
F	Textnummer oder in Verbindung mit ".." eine Formatanweisung
X	Spalte der Anzeige
Y	Zeile der Anzeige
Z	Zeichengröße für die Anzeige

Erläuterung

Die G253 ist eine Funktion zur Anzeige eines Textes oder einer Fehlermeldung an einer vordefinierten Stelle und vordefinierter Größe (applikationsabhängig; Standard: Statuszeile über Softkeys in normaler Größe) auf dem Bedienrechner. Mit X ist eine andere Spalten- und mit Y eine andere Zeilenposition für die Textausgabe wählbar. Zusätzlich ist unter Z eine alternative Zeichengröße vorgebbar.

Die erste und einfachste Verwendungsform der Funktion ist, den zur Ausgabe bestimmten Text in Kommentarzeichen "(", ")" in die gleiche Zeile wie die G253 einzugeben. Zu beachten ist, dass hinter dem Kommentar keine Parameter mehr bis zum Ende der Zeile ausgewertet werden.

Die 2. Möglichkeit der Textanzeige ist die Angabe einer Textnummer unter F. Der Bedienrechner bringt in diesem Falle den unter der angegebenen Nummer abgelegten Text zur Anzeige.

Die 3. Variante ermöglicht zusätzlich den Einbau von bis zu drei numerischen Werten in den Text, die unter A, B und C angegeben werden. Hierzu ist die Angabe des Ausgabetextes unter F= in Anführungszeichen ("...") zu programmieren. Die Formatanweisung darf, im Gegensatz zum Kommentar, an beliebiger Stelle im Satz stehen. Ein zusätzlicher Kommentar wird bei Vorhandensein der Formatanweisung nur noch als Kommentar interpretiert.

Hinweis!

Die G253 ist nur im ersten NC-Kanal zulässig. Im Satzvorlauf wird die G253 ignoriert.

Die Position und das Format der einzufügenden Parameter sind an der gewünschten Stelle im Textstring gemäß den nachfolgenden Regeln zu kennzeichnen.

%[Flags][Breite][Genauigkeit]Typ

Flags	
<nichts>	rechtsbündig, führende Leerzeichen/Nullen
-	linksbündig, nachfolgende Nullen
+	Vorzeichen immer ausgeben (auch +)
” ”	nur neg. Vorzeichen ausgeben
#	bei le,lE,lf : Dezimalpunkt immer ausgeben
bei lg, lG:	Dezimalpunkt immer ausgeben aber ohne folgende Nullen

Breite	
0n	min n-Stellen, Nullen vorangestellt
n	min n-Stellen, ” ” vorangestellt

Genauigkeit	
<nichts>	6 Stellen bei le,lE,lf
.0	e,E,f kein Dezimalpunkt ausgeben
.n	max. n-Nachkommastellen

Typ	
lf	Fließkomma [-]dddd.ddd
le	Fließkomma [-]d.ddd e[±]ddd
lg	wie le oder lf; je nach Genauigkeit automatische Umschaltung der Darstellung
lE	wie le, mit ”E” vor dem Exponenten
lG	wie lG, mit ”E” vor dem Exponenten
%	das Zeichen % soll angezeigt werden

Als Parameter sind mit indirekter Programmierung (A=P...) Parameterfeldindices unter A, B und C vorgebbar, wobei die alphabetische Reihenfolge auch die Reihenfolge bei der Konvertierung ist.

Ist die Adresse E programmiert, so wird eine nach dem unter E programmierten Wert abgestufte Fehlerbehandlung eingeleitet. Hierbei wird der im Kommentar oder unter F= verankerte Text als Fehlertext ausgegeben. Erlaubte Werte für E sind 101 ... 499. Die Hunderter-Stelle des unter E programmierten Wertes wird als ”Fehlergewichtung” interpretiert. Dem unter E angegebenen Wert wird 2000 hinzuaddiert, und als Fehlermeldung mit der Modulkennung 3 (Interpreter) am Bediengerät angezeigt.

Wenn E programmiert ist, werden die Angaben unter X, Y und Z ignoriert. Wenn E nicht programmiert ist, erfolgt keine Fehlerbehandlung.

3.3 Formelprozessor

Im NC-Programm ist außer der Programmierung über G-Funktionen auch die direkte Eingabe mathematischer Formeln möglich. Ein mathematischer Ausdruck wird durch ein ":" am Zeilenanfang gekennzeichnet. Eine Satznummer darf vorangestellt werden. Kommentare sind in einem Satz mit mathematischem Ausdruck zulässig unter Einschluss in geschweifte Klammern "{}".

Mit Hilfe des Formelprozessors können auch komplexe Ausdrücke zur Laufzeit der Steuerung berechnet werden. Als Operanden sind dabei sowohl Konstanten als auch Elemente aus dem Parameterfeld verwendbar und beliebig kombinierbar. Die Rechenoperationen werden durch vereinbarte Operationen gekennzeichnet. Die Verwendung von Klammern ist zulässig.

3.3.1 Rechenoperationen

Syntax	Operation	Beschreibung
x + y	x + y	Addition
x - y	x – y	Subtraktion
x * y	xy	Multiplikation
x / y	x / y	Division
x ^ y	x^y	Potenz
SQR(x)	x^2	Quadrat einer Zahl
SQRT(x)	$\sqrt{x}$	Quadratwurzel einer Zahl
HYPOT(x,y)	$\sqrt{x^2 + y^2}$	Betrag eines Vektors in der Ebene
HYPOT3(x,y,z)	$\sqrt{x^2 + y^2 + z^2}$	Betrag eines Vektors im Raum
FMOD(x,y)	rest(*x/y*)	Rest der aus Division x / y
SIN(x)	sin*x*	Sinus eines Winkels in Grad
COS(x)	cos*x*	Cosinus eines Winkels in Grad
TAN(x)	tan*x*	Tangens eines Winkels in Grad
ASIN(x)	arcsin*x*	Arcus Sinus, liefert einen Winkel zwischen -90° und +90°
ACOS(x)	arccos*x*	Arcus Cosinus, liefert einen Winkel zwischen 0° und 180°
ATAN(x)	arctan*x*	Arcus Tangens, liefert einen Winkel zwischen -90° und +90°
ATAN2(y,x)	arctan(*y/x*)	Arcus Tangens mit zwei Argumenten, liefert einen Winkel zwischen 0° und 360°
MOD360(x)	rest (*x*/360)	Winkel x modulo 360°
ABS(x)	|*x*|	Absolutwert einer Zahl (ohne Vorzeichen)
SIGN(x)	x/ x	Vorzeichen einer Zahl; +1 für x >= 0 und -1 für x < 0
ROUND(x)	round*x*	Auf- bzw. Abrunden einer Zahl auf ganzzahlige Werte
TRUNC(x)	trunc*x*	Liefert die Vorkommastellen einer Zahl
FRAC(x)	x-truncx	Liefert die Nachkommastellen einer Zahl
MIN(x,y)	min{*x,y*}	Liefert den kleineren Wert
MAX(x,y)	max{*x,y*}	Liefert den größeren Wert
RANDOM(x,y)	rand	Liefert Zufallszahlen zwischen x und y

Syntax	Operation	Beschreibung
AND(x,y)	$(x \neq 0) \wedge (y \neq 0)$	Logische Und-Verknüpfung
OR(x,y)	$(x \neq 0) \vee (y \neq 0)$	Logische Oder-Verknüpfung
XOR(x,y)	$(x \neq 0) \veebar (y \neq 0)$	Logische Exclusive-Oder-Verknüpfung
NOT(x)	x = 0	Logisches Nicht
BITAND(x,y)	$x \wedge y$	Bitweise Und-Verknüpfung
BITOR(x,y)	$x \vee y$	Bitweise Oder-Verknüpfung
BITXOR(x,y)	$x \veebar y$	Bitweise Exclusive-Oder-Verknüpfung
BITNOT(x)	$\overline{x}$	Bitweises Nicht
PI	π	Konstante 3,141592654
PI180	$\pi/180$	Konstante 0,01745329252

Die Rechenoperationen werden in der den mathematischen Regeln entsprechenden Reihenfolge bearbeitet. Zur Kenntlichmachung von Vorrangstellungen sind Klammern bis zu einer Schachtelungstiefe von 4 Ebenen erlaubt.

Beispiel

N10 : P1050 = sin(89.9 * P1051) / (34 + 91 * cos(P1052))
N20 : P1060 = atan2(P1,P0)
N30 : P1061 = hypot(P0,P1)

3.4 Satzerweiterungen

Die Sätze dürfen um eine oder mehrere zusätzliche Funktionen, die durch die Adressbuchstaben **H, M**, **P**, **Q**, **S** und **T** gekennzeichnet sind, erweitert werden. Die Funktionen dürfen einzeln oder zu mehreren in einem Satz programmiert werden. Sie dürfen auch alleine in einem Satz stehen.

Wenn mehrere Funktionen gleichzeitig angegeben sind, werden sie in der festen Reihenfolge **H, P**, **T**, **S**, **M**, **Q** bearbeitet.

3.4.1 Parameterzuweisung P

In jedem Satz können einfache Parameterzuweisungen direkt programmiert werden, ohne G-Funktion oder Formelprozessor. Für komplexe Zuweisungen, die Rechenoperationen enthalten, muss jedoch der Formelprozessor verwendet werden.

Die Parameterfeldzuweisung darf innerhalb eines Satzes auch mehrfach programmiert sein. Dies kann unter anderem bei Unterprogrammaufrufen ausgenutzt werden, um mehrere Parameter zu übergeben. Die Parameterfeldzuweisung wird immer als erste Operation in einem Satz ausgeführt, unabhängig von der Stelle im Satz, an der sie programmiert ist.

Mögliche Zuweisungsvarianten sind:

P2000=17: Die Zahl 17 wird im Parameter 2000 abgelegt.

P2000=P0: Der Inhalt von P0 wird dem Parameter 2000 zugewiesen.

PP1500=3.5: Dem Parameter, dessen Index in P1500 angegeben ist, wird der Wert 3,5 zugewiesen.

P2000=PP10: P2000 wird der Inhalt des Parameters zugewiesen, dessen Index in P10 eingetragen ist.

PP10=PPP11: Dem Parameter, dessen Index in P10 angezeigt ist, wird der Wert des Parameterfeldes zugewiesen, dessen Index in dem Parameterfeld steht, dessen Index in P11 abgelegt ist.

Beispiel

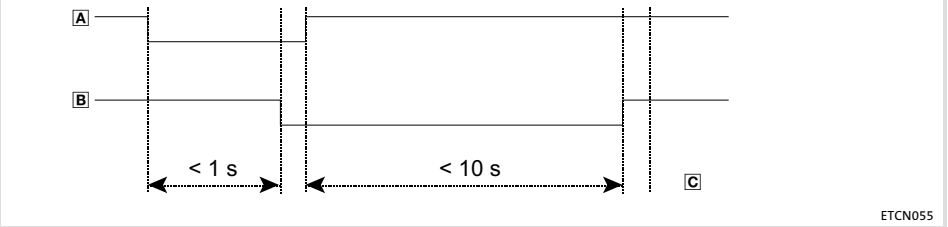
```
N10 G0 X3.14 P4000=0 P2000=P0
P3000=PP200 P3100=123.2 P3333=P100
G22 L7999 P2000=1 P2001=P1
```

3.4.2 H-Funktionen

H-Funktionen sind für die Änderung von Technologie-Parametern vorgesehen. Die bei H programmierte Nummer wird an die SPS übergeben und muss dort ausgewertet werden.

Die H-Funktion wird immer als erste Funktion im Satz ausgewertet. H-Funktionen sind generell zeitsynchronisiert, d. h. die nachfolgende Funktion wird erst interpretiert, wird die Übergabe von der SPS quittiert wurde.

Die Übergabe von H-Funktionen an die SPS erfolgt mit Handshake per Strobe und Quittung. Mit der negativen Flanke des Strobe signalisiert der NC der SPS eine neue H-Funktion.



A Strobe
B Quittung
C Nächster Satz

Sollte die Übertragung der Technologieparameter länger als 10 Sekunden dauern, muss die SPS zusätzlich die Einlesefreigabe wegnehmen, sonst meldet die Steuerung die Fehlermeldung ”Time Out bei der H-Funktionsübergabe an die SPS”.

3.4.3 M-Funktionen

M-Funktionen werden an die SPS gesendet. Diese müssen dort ausgewertet und die gewünschte Funktion ausgelöst werden. Es wird unterschieden zwischen ”synchronisierten” und ”nicht synchronisierten” M-Funktionen.

Synchronisiert sind M-Funktionen mit den Nummern 0 ... 499. Werden diese an die SPS gesendet, so wird die Abarbeitung des NC-Programms bis zum Eintreffen der SPS-Quittung und Setzen der Einlesefreigabe unterbrochen.

Bei nicht synchronisierten M-Funktionen (>= 500) wird nach der Ausgabe anschließend sofort der nächste Satz des NC-Programms ausgeführt ohne auf eine Quittung seitens der SPS zu warten. Dabei ist zu beachten, dass es auch bei ausgeschaltetem Genauhalt zu einem Geschwindigkeitseinbruch zwischen zwei aufeinanderfolgenden Verfahrsätzen kommt, falls mehrere M-Funktion hintereinander ohne eine Verfahrbewegung programmiert sind.

Folgende M-Funktionen werden auch im NC-Rechner ausgewertet:

M30: Programmende

M14: Schaltet von die Aufsummierung der Bahnlänge in P551 ein (Kontur)

M15, M16, M21: Schaltet von die Aufsummierung der Bahnlänge in P551 aus (Leerfahrt)

M48: Geschwindigkeitsüberlagerung wirksam (Override ein)

M49: Geschwindigkeitsüberlagerung unwirksam (Override aus)

M1014: Wie M14, jedoch als asynchrone M-Funktion

M1015: Wie M15, jedoch als asynchrone M-Funktion

M1048: Wie M48, jedoch als asynchrone M-Funktion

M1049: Wie M49, jedoch als asynchrone M-Funktion

Es besteht die Möglichkeit, durch eine M-Funktion zusätzlich einen Unterprogrammaufruf zu initiieren. Dazu müssen in den Maschinenkonstanten, unter dem Bezeichner **MK_MFKT_UPR_TABELLE**, alle so zu expandierenden M-Funktionen eingetragen werden.

Kommt die Steuerung im Zuge der Programmabarbeitung zu einer entsprechenden M-Funktion, so wird sie, nach(!) Abarbeitung aller anderen Informationen aus diesem Satz, einen Unterprogrammaufruf mit der Programmnummer ”9000 + M-Funktionsnummer” erzeugen.

Es können für diese Art des Unterprogrammaufrufes M-Funktionsnummern von 1 bis 255 genutzt werden.

Die M-Funktion selbst wird ganz regulär zur SPS übertragen. Ist in dem so aufgerufenen Unterprogramm die gleiche M-Funktion programmiert, führt dieser Umstand nicht zu einem erneuten Unterprogrammaufruf.

Stop!

Wird in dem Unterprogramm 9000(+M-Funktionsnummer) ein anderes Unterprogramm aufgerufen, das seinerseits wieder die auslösende M-Funktion beinhaltet, kommt es zu einer Aufrufrekursion und schließlich zu einer Fehlermeldung ”Programmstack overflow”

Dieser Unterprogramm-Mechanismus ist nicht nur in der automatischen Programmabarbeitung aktiv. Die gleiche M-Funktion wird auch als Einzelsatz in einem Unterprogrammaufruf expandiert.

Beispiel

```
N10 G0 X300 M14
N20 G2....
```

Nachdem die Steuerung die X-Achse auf den Wert 300 verfahren hat, und die Information über die M14 an die SPS übertragen wurde, wird das Unterprogramm %9014 ausgeführt. Erst nach Abschluss des Unterprogramms %9014 wird mit dem Satz N20 fortgefahren.

Die Programmsequenz des Beispiels ist funktional identisch mit

```
N10 G0 X300 M14
G22 L9014
N20 G2....
```

3.4.4 Q-Funktionen

Unter der Adresse ”Q” sind Zustände von CAN-E/A-Baugruppen programmierbar. Die Q-Funktionen werden auch als ”schnelle Eingänge” bezeichnet, da sie direkt im Grobinterpolator, ohne Umweg über die SPS in das NC-Programm gegeben werden.

Da bei den digitalen Ausgängen nur CAN-Ausgänge zur Verfügung stehen, sind diese immer an den SPS-Zyklus gebunden.

Die SPS kann den Ausgangszustand der betroffenen Ausgänge nicht mehr verändern, bis die Freigabe von der SPS rückgesetzt wird. Sie kann jedoch unabhängig davon den aktuellen Ausgangszustand jederzeit lesen.

”Schnelle Ausgänge” werden von der Steuerung nach Programmende oder Abbruch nicht automatisch rückgesetzt. Wenn notwendig, muss dies von der SPS durch Wegnehmen der Freigabe und Rücksetzen der Ausgänge erfolgen.

Weitere Informationen zur Freigabe der “schnellen Ein-/Ausgänge” entnehmen Sie bitte der Dokumentation MC und NC Software Handbuch.

Q-Anweisungen dürfen innerhalb eines Satzes auch mehrfach programmiert sein.

Beispiel

Q0=0 Q14=0 Q15=1	Zustände der Ausgänge initialisieren
G60 X0	Genauhalt ausschalten
G0 X159 Y31 Q0=1	Position 159 in X und 31 in Y anfahren, Ausgang 0 auf 1 setzen und
G1 X163 Y77	ohne Stop den nächsten Satz ausführen
G1 Y121 Q14=1 Q15=0	Weiterfahren und am Ende des Satzes, Ausgang 14 auf 1 setzen und
G1 Y154	Ausgang 15 auf 0 setzen

3.4.5 S-Funktionen

Der Buchstabe S im DIN Satz steht immer für die Angabe des aktuellen Werkstückkoordinatensystems, auf das sich nachfolgende Positionsangaben beziehen. Unter einem Koordinatensystem versteht die Steuerung die Definition eines Nullpunkt-Offsets pro Achse (bezogen auf den Nullpunkt in S0), wobei S0 auf den Maschinennullpunkt (Referenzpunkt + Grundoffset) bezogen ist.

In ETCXC stehen jeweils 100 solcher Werkstück-Koordinatensysteme zur Verfügung. Diese werden mit S0 ... S99 angewählt.

Ist in der Maschinenkonstante MK_NULLPUNKTE_SPEICHERN eine 1 eingetragen, wird das aktuell eingeschaltete S-Koordinatensystem und die Nullpunkt-Offsets aller Koordinatensysteme gespeichert und nach dem Einschalten der Steuerung wiederhergestellt. Andernfalls befindet sich die Steuerung nach dem Einschalten in S0 und die Offsets aller Koordinatensysteme sind Null.

Für alle Werkstück-Koordinatensysteme kann mit den Funktionen G92, G93 und G193 der Nullpunkt verschoben bzw. gesetzt werden. S0 kann mit diesen Funktionen nur verschoben werden, wenn die Maschinenkonstante MK_S0T0_VERSATZ_ERLAUBT auf 1 gesetzt ist. Bei der Verschiebung von S0 ist in diesem Fall zu beachten, dass alle anderen Koordinatensysteme gleichfalls verschoben werden!

Die Funktion G195 dient speziell dazu, alle Koordinatensysteme zu verschieben in dem der Offset von S0 verschoben wird. Sie ist auch erlaubt, wenn die oben genannte Maschinenkonstante auf 0 gesetzt ist. Allerdings ist dann der Offset von S0 nicht in S0 aber in allen anderen Koordinatensystemen wirksam.

Mit der Funktion G54 kann der Nullpunkt des aktuellen S-Koordinatensystems temporär verschoben werden. Die Verschiebung bleibt solange aktiv, bis sie mit G53 aufgehoben oder das Programm beendet wird. Die Verschiebung durch G54 wird nicht remanent gespeichert.

Beim Umschalten auf ein anderes Koordinatensystem werden die momentanen Ist-Positionen in S_{alt} mit den Nullpunkt-Offsets des neuen Systems (S_{neu}) verrechnet und angezeigt.

Die Offsets der Werkstück-Koordinatensysteme können über das ETC-MMI in die Steuerung geladen bzw. von der Steuerung in den PC übertragen werden.

Beispiel

G0 X0 Y0 S0	Im Koordinatensystem S0 auf die Position X0, Y0 fahren.
G92 X0.1 S1	Umschalten ins Koordinatensystem S1 und Verschieben des Nullpunkts der X-Achse um 0.1 mm.
G1 X100 F2500	Kontur ist S1 Abfahren
M6 S=P711	Umschalten in das Koordinatensystem, dessen Nummer im Parameter P711 eingetragen ist und anschließend die M-Funktion M6 ausgeben.

3.4.6 T-Funktionen

Der Buchstabe T in DIN Satz steht immer für die Angabe des aktuellen Werkzeugkoordinatensystems, auf das sich nachfolgende Positionsangaben beziehen. Unter einem Koordinatensystem versteht die Steuerung die Definition eines Nullpunkt-Offsets pro Achse.

In ETCXC stehen jeweils 100 solcher Werkzeug-Koordinatensysteme zur Verfügung, die mit T0 ... T99 angewählt werden.

Zusätzlich zu den Offsets der Achsen ist jedem T-Koordinatensystem eine Tabelle von 30 Werkzeugdaten zugeordnet, die im Parameterfeld ab P700 für das aktuelle T eingeblendet sind. Diese Werkzeugdaten können innerhalb einer Werkzeugverwaltung verwendet werden, um werkzeugspezifische Parameter zu speichern. Diese werden bei der Werkzeugradiuskompensation verwendet.

Ist in der Maschinenkonstante MK_NULLPUNKTE_SPEICHERN eine 1 eingetragen, wird das aktuell eingeschaltete T-Koordinatensystem, die Nullpunkt-Offsets und die Werkzeugdaten aller Koordinatensysteme gespeichert und nach dem Einschalten der Steuerung restauriert. Andernfalls befindet sich die Steuerung nach dem Einschalten in T0 und die Offsets aller Koordinatensysteme und alle Werkzeugdaten sind Null.

In allen Werkzeug-Koordinatensystemen kann mit den Funktionen G92, G93 und G193 der Nullpunkt verschoben bzw. gesetzt werden.

Beim Umschalten auf ein anderes Koordinatensystem werden die momentanen Ist-Positionen in T_{alt} mit den Nullpunkt-Offsets des neuen Systems (T_{neu}) verrechnet und angezeigt.

Die Werkzeugdaten und die Offsets der Werkzeug-Koordinatensysteme können über das ETC-MMI in die Steuerung geladen bzw. von der Steuerung in den PC übertragen werden.

Ermittlung des Gesamt-Offsets der S- und T-Koordinatensysteme (P192ff)

MK_S0T0_VERSATZ_ERLAUBT = 0	
In S0: Offset = Tm	
In Sn: Offset = S0+Sn+Tm	mit n > 0
MK_S0T0_VERSATZ_ERLAUBT = 1	
in S0: Offset = S0+Tm	
In Sn: Offset = S0+Sn+Tm	mit n > 0

Beispiel

G193 X0 Y0 T1	Aktuelle Position der X- und Y-Achse in T1 auf Null setzen.
T5 S=P711	Umschalten auf T5 und danach das S-Koordinatensystem, welches in den Werkzeugdaten von T5 in P711 eingetragen ist, aktivieren.

3.5 Datenfelder

3.5.1 P-Feld

Für die Programmierung mit Variablen steht in der Steuerung ein Datenfeld zur Verfügung, das Parameter- oder P-Feld. Dieses Datenfeld enthält eine Reihe unterschiedlicher Daten:

- interne Daten des NC-Rechners

  Diese Daten werden durch die Bearbeitung eines Programms erzeugt und / oder spiegelt bestimmte Zustände des NC-Rechners wieder. Auf diese Daten kann nur lesend zugegriffen werden. Der Zugriff auf die meisten dieser Daten ist zeitsynchronisiert.

- Daten für Zyklen und Makros

  Dieser Bereich ist für die Verwendung in Zyklen und Makros reserviert. In diesem Bereich werden Datenübergaben vom NC-Rechner an die Zyklen vorgenommen. Darüber hinaus dient er als Variablenbereich für die Zyklenprogramme. Zyklen und Makros haben schreibenden und lesenden Zugriff auf diese Daten, Anwenderprogramme sollten nur lesend auf diesen Bereich zugreifen. Der Zugriff ist nicht zeitsynchronisiert.

- Daten für Anwenderprogramme

  Dieser Bereich ist für die freie Verwendung in Anwenderprogrammen vorgesehen. Diese haben schreibenden und lesenden Zugriff. Der Zugriff ist nicht zeitsynchronisiert. Die Bedeutung der Parameter ist programmabhängig.

Das Parameterfeld hat defaultmäßig eine Größe von 2048 Einträgen, diese kann mit der Maschinenkonstante MK_PFELD_GROESSE benutzerspezifisch vergrößert werden.

Aufteilung

0 1023	Interne Daten des NC-Rechners
1024 1499	Daten für Zyklen
1500 2047	Anwenderdaten
2048 MK_PFELD_GROESSE	Erweitertes P-Feld

Der Zugriff auf das Parameterfeld kann erfolgen durch

- G-Funktionen
- Arithmetikfunktionen (Formelprozessor)
- Nachrichten des MMI
- Nachrichten der PLC

Belegung der Systemparameter

Im Folgenden ist die Belegung des Bereichs des Parameterfeldes angegeben, in dem die internen Daten des NC-Rechners abgelegt sind. Die angegebenen Ziffern entsprechen den Parameternummern, unter denen die Variablen angesprochen werden.

Achsbezogene Werte sind immer in der Reihenfolge eingetragen, in der sie in der Maschinen-konstante MK_APPLACHSIDX definiert sind.

Bedeutung der Einheiten in den nachfolgenden Tabellen:

Einheit	Bedeutung
Inc	Incremente
°	Grad
Einh	Translatorische Wegeingabeeinheit, maschinenkonstantenabhängig (Default = mm)
mm	Millimeter
m	Meter
h	Stunde
min	Minute
s	Sekunde
ms	Millisekunde
GIT	Grobinterpolationstakt
MK	Einheit ist abhängig von den Maschinenkonstanten
%	Der Parameter ist in Prozent angegeben
---	Der Parameter hat keine Einheit

Bei einigen Parametern ist die Einheit davon abhängig, ob der Parameter zu einer Linearachse oder einer Rotationsachse gehört. Bei Linearachsen ist die Eingabeeinheit zusätzlich vom Wert in den Maschinenkonstanten MK_METRISCH, MK_CONST_REL_MM und MK_CONST_REL_INCH abhängig. Für diese Parameter gilt die folgende erweiterte Tabelle.

Einheit	Bedeutung bei Linearachsen	Bedeutung bei Rotationsachsen
mm *	Eingabeeinheit	Grad
mm/min *	Eingabeeinheiten/Minute	Umdrehungen/Minute
m/s^2 *	$Meter/Sekunde^2$	$Umdrehungen/Sekunde^2$
Inc/mm *	Incremente/Eingabeeinheit	Incremente/Grad

Ein Kreuz im Feld ”Sync” zeigt an, dass der Zugriff auf den entsprechenden Parameter zeitsynchronisiert wird, d. h. dass der Inhalt des Parameters innerhalb eines NC-Programms erst gelesen wird, wenn der Satz davor ausgeführt wurde. Solange wird auch die weitere Interpretation des NC-Programms angehalten.

Achs-spezifische Parameter

Index	Bedeutung	Sync	Einheit
0 ... 15	NC-Ist-Position der Achsen in Bezug auf das aktuelle Koordinatensystem	x	mm *
16 ... 31	Summe von modaler Ist-Position und NC-Ist-Position der Achsen (P160ff+P0ff)	x	mm *
32 ... 47	NC-Zielposition der Achsen in Bezug auf das aktuelle Koordinatensystem	x	mm *
64 ... 79	Nullpunkt-Offset von Koordinatensystem S0	x	mm *

Index	Bedeutung	Sync	Einheit
80 ... 95	Nullpunkt-Offset von Koordinatensystem Sn (-> P586) von Kanal N	x	mm *
96 ... 111	Nullpunkt-Offset von Koordinatensystem Sn (-> P554) von Kanal 0	x	mm *
112 ... 127	Nullpunkt-Offset von Koordinatensystem Tn (-> P585) von Kanal N	x	mm *
128 ... 143	Nullpunkt-Offset von Koordinatensystem Tn (-> P553) von Kanal 0	x	mm *
144 ... 159	Modale Zielposition der Achsen in Bezug auf das aktuelle Koordinatensystem	x	mm *
160 ... 175	Modale Ist-Position der Achsen in Bezug auf das aktuelle Koordinatensystem	x	mm *
176 ... 191	Modale Zielpositionsvorgabe in Bezug auf das aktuelle Koordinatensystem	x	mm *
192 ... 207	NC-Gesamt-Offset der Achsen von S0+Sn+Tn (kanal-unabhängig)	x	mm *
208 ... 223	Modale Geschwindigkeitsvorgabe der Achsen	x	mm/min *
224 ... 239	Echte Ist-Position der Achsen bezogen auf den Grundoffset	x	Inc
240 ... 255	Modale Beschleunigungsvorgabe der Achsen	x	m/s^2 *
256 ... 271	Schleppabstand der Achsen	x	Inc
272 ... 287	Modale Ist-Geschwindigkeit der Achsen	x	mm/min *
304 ... 319	Negative Software-Verfahrgrenzen der Achsen bezogen auf den Grundoffset	x	mm *
320 ... 335	Echte Ist-Position der Achsen bezogen auf den Grundoffset	x	mm *
336 ... 351	Positive Software-Verfahrgrenzen der Achsen bezogen auf den Grundoffset	x	mm *
352 ... 367	Gespeicherte Ist-Positionen der Achsen bezogen auf den Grundoffset, diese werden durch ein externes Triggersignal zeitgleich gespeichert	x	mm *
394 ... 409	Zustände der Eingangssignale der Achsen (Endschalter, Referenznocken,...) Bit0: positiver Endschalter Bit1: negativer Endschalter Bit2: Referenznocken Bit3: Reserveeingang	x	---
410 ... 425	Zustände der Eingangssignale der Slave-Achsen bei Synchronachsen. Bit0: positiver Endschalter Bit1: negativer Endschalter Bit2: Referenznocken Bit3: Reserveeingang	x	---
426 ... 441	Synchronabstand zwischen Master- und Slave-Achsen bei Synchronachsen	x	mm *
442 ... 457	Min. Synchronabstand, ermittelt über 200 Feininterpolationstakte	x	mm *
458 ... 473	Max. Synchronabstand, ermittelt über 200 Feininterpolationstakte	x	mm *
480 ... 495	Umrechnungsfaktor von Eingabeeinheiten in Inkremente	x	Inc/mm *
496 ... 511	Grundoffset aus den MK's (MK_GRUNDOFFSET)	x	mm *
640 ... 655	Vorgabegeschwindigkeit der Achsen zur Verwendung in Zyklen, wird auf den Wert der Maximalgeschwindigkeit in MK_VMAX initialisiert	x	mm/min *
928 ... 943	Zusätzlicher Grundoffset der Achsen (G194)	x	mm *

Index	Bedeutung	Sync	Einheit
944 ... 959	Lagesollpositionenen der Achsen inclusive aller Transformationen und Koordinatensystemoffsets	x	mm *
960 ... 975	Streckungsfaktoren der Achsen (G75)		---
992 ... 1007	Programmierte Zielposition aus dem letzten interpretierten DIN-Satz		mm *

Parameter zur Programmverwaltung

Index	Bedeutung	Sync	Einheit
512,513	Programmnummer und Satznummer des startenden NC-Programm im Startmode 0 (MMI)	x	---
514,515	Programmnummer und Satznummer des startenden NC-Programm im Startmode 1 (SPS)	x	---
516,517	Programmnummer und Satznummer des startenden NC-Programm im Startmode 2 (ZYK)	x	---
518,519	Programmnummer und Satznummer des startenden NC-Programm im Startmode 3 (APP1)	x	---
520,521	Programmnummer und Satznummer des startenden NC-Programm im Startmode 4 (APP2)	x	---
522,523	Programmnummer und Satznummer des startenden NC-Programm im Startmode 5 (APP3)	x	---
524,525	Programmnummer und Satznummer des startenden NC-Programm im Startmode 6 (APP4)	x	---
526,527	Programmnummer und Satznummer des startenden NC-Programm im Startmode 7 (APP5)	x	---
528	Programmnummer als Triggerbedingung für die Funktion Satzvorlauf	x	---
529	Satznummer als Triggerbedingung für die Funktion Satzvorlauf	x	---
530	Logische Satznummer als Triggerbedingung für die Funktion Satzvorlauf	x	---
531	M-Funktion als Triggerbedingung für die Funktion Satzvorlauf	x	---
532	Anzahl der Schleifendurchläufe als Triggerbedingung für die Funktion Satzvorlauf	x	---
533	Bahnlänge, um die in den Zielsatz innerhalb der Funktion Satzvorlauf hineinpositioniert werden soll	x	---
534	Zeilennummer als Triggerbedingung für die Funktion Satzvorlauf	x	
535	Werkzeugnummer als Triggerbedingung für die Funktion Satzvorlauf	x	
536	Geometrienummer als Triggerbedingung für die Funktion Satzvorlauf	x	
537	Position im Zielsatz als Maß zwischen 0 und 1 für der Funktion Satzvorlauf. 0: Satzanfang 1: Satzende	x	
540	Bitcodierte Anzeige, die signalisiert, ob die Funktion Oszillieren (G36) für eine Achse eingeschaltet ist	x	---
541	Zeigt an ob die gepufferten Positionen in P352 ... 367 gültig sind, sie sind gültig, wenn Triggersignal erkannt wurde. Dieser Parameter muss erst auf 0 zurückgesetzt werden, um ein erneutes Latchen der Positionen zu ermöglichen.	X	---
544	Nummer des NC-Kanals für den die nachfolgenden Daten gültig sind.	x	---
545	Aktuelle Programmnummer	x	---
546	Aktuelle Satznummer	x	---

Index	Bedeutung	Sync	Einheit
547	Aktuelle logische Satznummer	x	---
551	Summe aller abgefahrenen Konturstücke ohne Leerfahrten seit dem Programmstart	x	---
552	Zurückgelegte Bahnlänge im aktuellen Satz	x	---
553	Aktuelles Werkzeugkoordinatensystem Tn	x	---
554	Aktuelles Werkstückkoordinatensystem Sn	x	---
555	Aktuelle Bahngeschwindigkeit	x	mm/GIT *
556	Aktueller Bearbeitungszustand von Kanal 0 zu Diagnosezwecken 0: Idle 1: Run 2: Bremsen-Abbruch 3: Abbruch-Wait-Quit 4: Abbruch 5: Bremsen-Unterbrechen 6: Unterbrechen 7: Bremsen-Satzsprung 8: Satzsprung 9: Bremsen-Interrupt 10: Interrupt 11: Bremsen-Fehler 12: Fehler 13: Ausgleichsfahrt	x	
557	Aktuelle Schrittweite, z. B. Stichlänge beim Nähen	x	mm *
558	Aktueller Interpreterzustand 0: kein Programm aktiv 1: Einzelfunktion aktiv 2: Programm aktiv 3: Interruptprogramm aktiv 4: Satzvorlauf aktiv	x	---
559	Bitcodierter Zustand modaler Funktionen Bit0: 0=Streckenbetrieb (G30), 1=Bahnbetrieb (G31) Bit1: 0=Leerfahrt (M15,M16,M21), 1=Kontur (M14) Bit2: 0=Eventhandler disabled (G132), 1=Eventhandler enabled (G133) Bit3: 0=Override enabled (M48), 1=Override disabled (M49) Bit4: 1=G114 aktiv Bit5: 1=G231 aktiv	x	---
560	Aktuelle Zeilennummer (nicht Satznummer) im NC-Programm	x	---
561	Aktuelle Solldrehzahl, z. B. für eine Spindel. Entspricht dem bei der Adresse E im DIN-Satz programmierten Wert (G0-G3).	x	---
562	Nummer der M-Funktion, durch die der letzte Unterprogrammaufruf ausgelöst wurde. Zur Verwendung in Zyklen, die durch M-Funktionen aufgerufen werden (MK_MFKT_UPR_TABELLE).		---
563	Letzte ausgegebene H-Funktionsnummer	x	---
564	Aktuelle Anzahl der vorverarbeiteten Sätze im Satzvorlaufpuffer		---
565	Geometriezähler für Satzvortschrittsanzeige	x	---
566	Fortschrittsanzeige für aktuellen Verfahrsatz: 0=Satzanfang, 1=Satzende	x	---
567	Effizienz des Geometriefilters G200 in %	x	%
576	Nummer des NC-Kanals, für den die nachfolgenden Daten gültig sind.	x	---
577	Aktuelle Programmnummer	x	---
578	Aktuelle Satznummer	x	---

Index	Bedeutung	Sync	Einheit
579	Aktuelle logische Satznummer	x	---
583	Summe aller abgefahrenen Konturstücke ohne Leerfahrten seit dem Programmstart	x	---
584	Zurückgelegte Bahnlänge im aktuellen Satz	x	---
585	Aktuelles Werkzeugkoordinatensystem Tn	x	---
586	Aktuelles Werkstückkoordinatensystem Sn	x	---
587	Aktuelle Bahngeschwindigkeit	x	mm/GIT *
588	Aktueller Bearbeitungszustand von Kanal N zu Diagnosezwecken 0: Idle 1: Run 2: Bremsen-Abbruch 3: Abbruch-Wait-Quit 4: Abbruch 5: Bremsen-Unterbrechen 6: Unterbrechen 7: Bremsen-Satzsprung 8: Satzsprung 9: Bremsen-Interrupt 10: Interrupt 11: Bremsen-Fehler 12: Fehler 13: Ausgleichsfahrt	x	
589	Aktuelle Schrittweite, z. B. Stichlänge beim Nähen	x	mm *
590	Aktueller Interpreterzustand 0: kein Programm aktiv 1: Einzelfunktion aktiv 2: Programm aktiv 3: Interruptprogramm aktiv	x	---
591	Bitcodierter Zustand modaler Funktionen Bit0: 0=Streckenbetrieb (G30), 1=Bahnbetrieb (G31) Bit1: 0=Leerfahrt (M15,M16,M21), 1=Kontur (M14) Bit2: 0=Eventhandler disabled (G132), 1=Eventhandler enabled (G133) Bit3: 0=Override enabled (M48), 1=Override disabled (M49)	x	---
592	Aktuelle Zeilennummer (nicht Satznummer) im NC-Programm	x	---
593	Aktuelle Solldrehzahl, z. B. für eine Spindel. Entspricht dem bei der Adresse E im DIN-Satz programmierten Wert (G0-G3).	x	---
594	Nummer der M-Funktion, durch die der letzte Unterprogrammaufruf ausgelöst wurde. Zur Verwendung in Zyklen, die durch M-Funktionen aufgerufen werden (MK_MFKT_UPR_TABELLE).		---
595	Letzte ausgegebene H-Funktionsnummer	x	---
596	Aktuelle Anzahl der vorverarbeiteten Sätze im Satzvorlaufpuffer		---
597	Geometriezähler für Satzfortschrittsanzeige	x	---
598	Fortschrittsanzeige für aktuellen Verfahrsatz: 0: Satzanfang 1: Satzende	x	---
599	Effizienz des Geometriefilters G200 in %	x	%
610	Größe des freien Speichers zur Verwaltung von NC-Programmen		Byte
611	Größe des freien Speichers zur Verwaltung von symbolischen Programmnummern		Byte
612	Anzahl der anstehenden und noch nicht quittierten Fehlermeldungen		

Index	Bedeutung	Sync	Einheit
613	Wert von MK_S_VERSATZSPERRE aus den Maschinenkonstanten		
666 ... 669	Aktuelle Werte von Override 0 ... 3		
670	Maximalwert der Beschleunigungsrampe auf der Bahn, siehe G201		m/s^2
671	Maximalwert der Bremsrampe auf der Bahn, siehe G201		m/s^2
672	Geschwindigkeitsvorwahl für modales Fahren	x	Einh/min
673	Schrittweitenvorgabe für Schrittfahren		Einh
674	Umrechnungsfaktor von Eingabeeinheiten in mm		mm/Einh
675	Umrechnungsfaktor von m/min nach Eingabeeinheiten/GIT	x	Einh/GIT m/min
676	Umrechnungsfaktor von Eingabeeinheiten/min in Eingabeeinheiten/GIT	x	min/GIT
677	Umrechnungsfaktor von m/s^2 in Eingabeeinheiten/GIT^2	x	Einh/GIT^2 m/s^2
678	Umrechnungsfaktor von Eingabeeinheiten/s^2 in Eingabeeinheiten/GIT^2	x	s^2/GIT^2

Technologiespezifische Parameter

Index	Bedeutung	Sync	Einheit
288 ... 303	Analoge Messwerte von Kanal 0 ... 15 (-> MK_MESS_AUFLOESUNG)	x	MK
368 ... 373	Durch Messzyklen ermittelte Achspositionen für eine Grunddrehung	x	mm *
374 ... 376	Grunddrehwinkel A, B, C in der YZ, ZX und XY-Ebene	x	°
685	Gewindesteigung (G33)	x	mm *
686	Gewindesteigungskorrektur (G33)	x	mm *
687	Gewindezylinderkorrektur (G33)	x	mm *
688	Gewindemessabstand (G33)	x	mm *
690	Radius im polaren Maschinenkoordinatensystem (MK_MASCH_POL...)	x	mm *
691	Winkel im polaren Maschinenkoordinatensystem	x	°
692	Winkel der C-Achse im polaren Koordinatensystem	x	°
696	Winkel der schräggestellten X-Achse aus MK_X_WINKEL	x	°
697	Sinus des Winkels der schräggestellten X-Achse	x	---
698	Cosinus des Winkels der schräggestellten X-Achse	x	---
699	Tangens des Winkels der schräggestellten X-Achse	x	---
700 ... 725	Werkzeugdaten (Tn) von Kanal 0, P700 = Werkzeugradius [mm]	x	---
730 ... 755	Werkzeugdaten von Kanal N (-> P576) P730 = Werkzeugradius [mm]	x	---
760 .. 769	Technologiedaten aus MK_TECHNOLOGIEDATEN1		---
770 ... 779	Technologiedaten aus MK_TECHNOLOGIEDATEN2		---
780 ... 789	Technologiedaten aus MK_TECHNOLOGIEDATEN3		---
790 ... 799	Technologiedaten aus MK_TECHNOLOGIEDATEN4		---

Parameter zur Zeiterfassung

Index	Bedeutung	Sync	Einheit
377	Gesamtlaufzeit seit Start des NC-Programms	x	GIT
378	Summe der Fahrzeiten seit Start des NC-Programms	x	GIT
379	Summe der Wartezeiten für M-Funktionen seit Start des NC-Programms	x	GIT
380	Zeit seit dem Einschalten der Steuerung	x	s
381	Zeit seit dem Einschalten der Steuerung	x	min
382	Zeit seit dem Einschalten der Steuerung	x	h
389	Summe der Verweilzeiten seit Start des NC-Programms	x	ms
390	Gesamtlaufzeit seit dem Start des NC-Programms	x	ms
391	Summe der Fahrzeiten seit Start des NC-Programms	x	ms
392	Summe der Wartezeiten für M-Funktionen seit Start des NC-Programms	x	ms
393	Summe der Wartezeiten durch Q-Tab und V-Tab-Vergleiche seit dem Start des NC-Programms	x	ms

Technologie-spezifische Anwenderparameter

Die Parameter im Bereich von P1024 bis P1099 sind für technologiespezifische Funktionen vorgesehen und nur für die jeweilige Technologie von Bedeutung. Bei allen hier nicht aufgeführten Technologien stehen die Parameter P1024 ... P1099 dem Anwender frei zur Verfügung.

Stoff-Schneiden

Index	Bedeutung	Sync	Einheit
1024	Overcut-Länge		mm *
1025	Messerlänge		mm *
1026	Gesamtmesserabrieb		mm *
1027	Mindestlänge einer Leerfahrt zum Anheben des Niederhalters		mm *
1028	Kopie aus MK_KONTURWINKEL2		°
1030	Advance-Before-Plunge		mm *
1032	Min. Messeroszillationsgeschwindigkeit (G97: Spindeltyp 2)		1/min
1033	Bahngeschwindigkeit für min. Messeroszillation		mm/min *
1034	Max. Messeroszillationsgeschwindigkeit		1/min
1035	Bahngeschwindigkeit für max. Messeroszillation		mm/min *
1036	Bahnwegfenster als Ausstechkriterium für das Messerschleifen bei der kundenspezifischen Vergleichsfunktion Typ 0 (G130 V0)		mm *

Styropor-Schneiden

Index	Bedeutung	Sync	Einheit
1040	Verweilzeit bei nichttangentialen (ungleich 180°) Konturübergängen		s
1041	Vorabschaltposition des Heizdrahtes in X		mm *
1042	Max. erreichte Position der X-Achse vor dem Stillstand von X		mm *
1043	Voreinschaltposition der X-Achse, bei der der Heizdraht eingeschaltet wird		mm *

Achspositionier-Handler für Handlingsaufgaben (G97 X10)

Index	Bedeutung	Sync	Einheit
1040	Sollgeschwindigkeit für die Achspositionierung		mm/min *
1041	Zielgeschwindigkeit bei Erreichen des Zielpunktes		mm/min *
1042	Soll-Verfahrweg nach erfolgter Antriggerung		mm *
1043	Akkumulierter Ist-Verfahrweg nach erfolgter Antriggerung		mm *
1044	Nummer des Eingangs der als Triggersignal verwendet werden soll (0 ... 23)		---
1045	Positionierzustand 0: Stillstand, 1: Suchen des Triggersignals, 2: Positionierung läuft, 3: Sollverfahrweg erreicht bzw. überschritten (P1041 != 0), 4: Sollverfahrweg erreicht, Stillstand (P1041 == 0)		---
1046	Beschleunigungs- und Bremsrampe		m/s^2 *
1047	Aktuelle Vorgabegeschwindigkeit an die Positionierachse		mm/min *

Nähen (G97 X5|X6)

Index	Bedeutung	Sync	Einheit
1050	Y-Korrektur-Wert der pro mm Vorschub der X-Achse auf die Position der Y-Achse aufaddiert wird. Nur bei Spindeltyp 6		---
1051	Anzahl der Stiche von Nähkopf 1 seit dem Einschalten der Steuerung		---
1052	Anzahl der Stiche von Nähkopf 2 seit dem Einschalten der Steuerung, nur bei Spindeltyp 5 und zwei Nähköpfen		---
1057	Drehzahlvorgabe von SPS an Spindelhandler für Drehzahlreduzierung beim Annähen.		1/min
1058	Bahnzielvorgabe für kundenspezifische Vergleichsoperation vom Typ 2		mm *
1059	Eckendrehzahl des Nähmotors in Prozent der Solldrehzahl, zur Drehzahlreduzierung bei nichttangentialen Satzübergängen		%

Freifahrhandler für Mess-Maschine

Index	Bedeutung	Sync	Einheit
1060	Wird gesetzt, wenn eine Antastung erfolgt ist. Kann im Antastzyklus als Bedingung für eine modale Verzweigung (G130) verwendet werden		---
1061	Anzahl der Fehltriggerungen zu Diagnosezwecken		---
1062	Antastweg, reserviert für Antastzyklus		mm *
1063	Antastwegtoleranz, reserviert für Antastzyklus		mm *
1064	Antastgeschwindigkeit, reserviert für Antastzyklus		mm/min *
1065	Bremsrampe zum Anhalten der Verfahrbewegung nach einer Antastung		m/s^2 *
1066	Freifahrgeschwindigkeit nach einer Antastung		mm/min *
1067	Freifahrbeschleunigung nach einer Antastung		m/s^2 *
1068	Freifahrweg relativ zur Antastposition		mm *
1069	Wartezeit nach Antasttriggerung zum Ausfiltern von Fehltriggerungen		ms
1070	Antaststatus 0: noch keine Antastung erfolgt 1: gültige Antastung erfolgt 2: Antastung könnte gültig sein, Entscheidung im nächsten GIT		
1071 ... 1086	Gespeicherte Antastpositionen, diese werden bei der pos. Flanke am Triggersignal übernommen		mm *

3.5.2 Q-Feld

Das Q-Feld ist ein bitweise orientiertes Feld von 64 Bit-Größe. Über das Q-Feld kann der Ablauf eines NC-Programms (z. B. G150) durch externe Ereignisse gesteuert werden. Die einzelnen Bits können von der SPS im Datenbaustein 1 beeinflusst werden. Die Belegung kann vom Anwender frei definiert werden.

Falls schnelle Eingänge konfiguriert sind (siehe MC und NC Software Handbuch), werden die zugehörigen Q-Bits nicht aus dem Datenbaustein 1 der SPS, sondern direkt von den entsprechenden Eingängen gelesen.

4 Maschinenkonstanten

Dieses Kapitel beschreibt die Maschinenkonstanten (MKs) der Steuerungen ETCPC und ETCHC. Einige der beschriebenen MK's stehen wegen der unterschiedlichen Hardware-Eigenschaften nicht in allen 2 Steuerungstypen zur Verfügung. Die Bezeichner, bei denen dies der Fall ist, sind mit einer entsprechenden Fußnote gekennzeichnet.

4.1 Grundlagen

MK's dienen der Anpassung der Steuerung an das konkrete Einsatzgebiet. Dieses sind zum einen die eigentliche Maschine mit ihren Achsen und zum anderen die Bereiche Technologie und Bedienphilosophie. Dem Endkunden und Anwender sollte deshalb die Beeinflussung der MK's nur eingeschränkt ermöglicht werden.

Am Ende dieses Kapitels ist eine vollständige Übersicht der MK's nebst Kommentierung abgedruckt. Diese wird mit jedem ETC-MMI unter dem Namen MUSTER.MK mitgeliefert, kann aber auch separat geliefert werden.

Hinweis!

Die Datei "MUSTER.MK" dient als Grundlage für Ihre spezifischen Anpassungen. Sie sollten sich deshalb eine Kopie der Datei erzeugen. Der Name der Datei spielt keine Rolle. Sie muss lediglich die Endung ".MK" aufweisen. Sinnvoll ist jedoch ein Name, der einen Bezug zur Maschine herstellt, z. B. die Maschinennummer

Die Datei kann auf zwei verschiedene Arten in die Steuerung übertragen werden. Entweder über ein ETC-MMI bei jedem Neustart der Steuerung oder mit einem beliebigen Terminal-Programm über die Monitorschnittstelle der Steuerung. Im ersten Fall wird die Datei auf der Festplatte des PC gespeichert. Der zweite Fall wird nur dann nötig, wenn kein PC-gestütztes MMI an die Steuerung angeschlossen ist.

Die MK's werden in der Steuerung in einem nicht-flüchtigen Speicher gehalten und werden dort bei jeder Änderung der MK's aktualisiert. Ab Werk ausgelieferte Steuerungen enthalten keine MK-Datei, daher kommt beim ersten Betrieb der Steuerung die Meldung "keine Maschinenkonstantendatei geladen".

Wenn Sie die MK-Datei anpassen, beachten Sie bitte, dass MK's, die nicht in der MK-Datei enthalten sind, nicht auf den Default-Wert zurückgesetzt werden, sondern den vorher eingestellten Wert behalten. Um eine bessere Übersicht zu haben, sollten jedoch alle MK's im MK-File beschrieben sein. Für einige MK's ist der mögliche Wertebereich eingeschränkt. Liegt der eingegebene Wert außerhalb dieses Wertebereiches, dann wird er auf diesen begrenzt.

Die MK's müssen genau mit ihrem Namen eingegeben werden, da sie sonst vom NC-Rechner nicht erkannt werden. Die ETC gibt beim Übertragen der MK's für jede nicht erkannte MK eine Fehlermeldung aus. Nach einer fehlerfreien Übertragung sind die MK's in der Maschine gültig. Grundsätzlich ist die Übertragung der MK's immer möglich. MK's, die die Speicherverteilung innerhalb der Steuerung beeinflussen, werden jedoch erst nach einem Neustart der Steuerung wirksam. Alle anderen MK's werden sofort nach der Übertragung wirksam bzw. spätestens, wenn die Steuerung bereit ist, ein Programm auszuführen.

Die MK's lassen sich in bestimmte zusammenhängende Bereiche einteilen. Die Reihenfolge ist jedoch beliebig.

4.2 Testeinstellungen

Die Testeinstellungen dienen dazu, die ETC ohne Maschine zu betreiben oder zu testen oder bestimmte Teile des Funktionsumfangs (SPS) auszuschalten.

4.2.1 MK_TEST_OHNEMECHANIK

Diese Maschinenkonstante dient dem Test der Steuerungsfunktionen, ohne dass eine Maschine angeschlossen werden muss. Dabei schaltet die Steuerung die Istwertgebereingänge in den Simulationsbetrieb. Stellgrößen werden weiterhin ausgegeben. Sollten bereits Antriebe angeschlossen sein, müssen die Servoverstärker abgeschaltet werden, da es sonst zu unerwünschten Verfahrbewegungen kommt!

Wert	Bedeutung
0	Antriebe und Wegmesssysteme müssen ordnungsgemäß angeschlossen sein (default).
1	Zum Testen von MMI-, SPS- und DIN-Programmen ohne angeschlossene Antriebe.

4.2.2 MK_SPS_DUMMY

Diese Maschinenkonstante schaltet die integrierte SPS ab oder an.

Wert	Bedeutung
0	Die geladene SPS wird automatisch ausgeführt (default). Falls beim Hochlauf der Steuerung diese MK auf 0 ist und keine SPS im Flash-PROM geladen ist, erfolgt eine Fehlermeldung.
1	Alle für den Betrieb der Steuerung notwendigen SPS-Freigaben werden gesetzt. M-Funktionen haben keine Wirkung, und alle Ein- und Ausgänge bleiben unbedient. Die geladene SPS wird nicht ausgeführt. Es erfolgt aber keine Fehlermeldung, wenn keine SPS geladen ist.
-1	Die SPS wird nicht ausgeführt und alle Freigaben rückgesetzt. Damit ist kein Programmstart und kein Verfahren von Achsen möglich. Es erfolgt keine Fehlermeldung, wenn keine SPS geladen ist.

Hinweis!

Beim Übertragen der MK's von einem ETC-MMI über die Dual-Port-RAM-Schnittstelle wird im Flash-PROM bei dieser MK immer -1 eingetragen, um beim nächsten Neustart der Steuerung zu verhindern, dass die SPS vorzeitig gestartet wird. Unabhängig davon wird im RAM der vom MMI übertragene Wert gehalten.

Nach Übertragung einer SPS durch die Programmieroberfläche **CodeSys** wird diese MK im RAM automatisch auf 0 gesetzt und die SPS gestartet. Der Wert im Flash-PROM bleibt unverändert.

4.3 Software-Konfiguration

4.3.1 MK_KUNDE

Diese Maschinenkonstante ist eine Stringkonstante, über die verschiedene kundenspezifische Erweiterungen innerhalb der Steuerung aktiviert werden. Dazu gehören technologiespezifische Korrekturmodule und M-Funktions-handler.

Wert	Bedeutung
""	Standard-WRK, keine Sonderbehandlungen von M-Funktionen (default).
"SCHNEIDEN"	WRK mit tangentialer Nachführung der C-Achse für Schneidtechnologien, Sonderbehandlung für M14, M15, M16 und M1014, M1015, M1016.
"SCHNEIDEN1"	wie "SCHNEIDEN", jedoch mit zusätzlichen Funktionen zur Verwaltung der Messerlänge am Konturanfang und Konturende, zusätzlich Sonderbehandlung für M46, M47, M50, M51 und M1046, M1047, M1050, M1051.
"SCHNEIDEN2"	WRK-Kompatibilitätsmodus mit tangentialer Nachführung der C-Achse für Schneidtechnologien ohne Sonderbehandlungen von M-Funktionen.
"SCHNEIDEN3"	wie "SCHNEIDEN2", jedoch wird zusätzlich bei G41/42 bzw. G40 automatisch ein rechtwinkliger Anfahr- bzw. Abfahrsatz eingefügt.
...	kundenspezifische Module, die hier nicht aufgeführt werden können.

4.3.2 MK_VOREINSTELLUNG

Diese Maschinenkonstante ist eine Stringkonstante, über die einige Voreinstellungen von G-Funktionen getroffen werden können. Folgende G-Funktionen dürfen innerhalb des Strings durch Semikolon getrennt angegeben werden:

Wert	Bedeutung
G0-G3 G100-G103	Modale G-Funktion setzen, optional mit Geschwindigkeitsangabe F Beispiel: AG1 F1000
G17 / G18 / G19	Auswahl der Hauptebene X-Y / Z-X / Y-Z
G26	Softwareendschalter der angegebenen Achsen aktivieren, Beispiel: AG26 X0 Y0
G30 / G31	Streckenbetrieb / Bahnbetrieb aktivieren
G60	Look-Ahead oder Genauhalt aktivieren, Beispiel: AG60 X0
G90 / G91	Absolutmaß / Kettenmaß für alle Achsen aktivieren
G112	Tangentiale Nachführung der angegebenen Achse (A,B,C,a,b,c) aktivieren, Beispiel: AG112 C1
G122	Wirkung der Verfahrtasten konfigurieren, Beispiel: AG122 X1

Der angegebene String muss in Anführungszeichen angegeben werden und darf maximal 80 Zeichen lang sein. Die konfigurierten Voreinstellungen werden beim Hochlauf der Steuerung nach dem Laden der MK sowie am Programmende und nach einem Programmabbruch automatisch aktiviert.

Beispiel

MK_VOREINSTELLUNG "G17; G1 F30000; G26 X0 Z0";

Weitere Informationen zu den G-Funktionen finden Sie in Kapitel "CNC-Programmierung".

4.3.3 MK_NCPROG_OHNE_KOMMENTARE

Diese Maschinenkonstante unterdrückt die Abspeicherung von Kommentaren und Leerzeichen in DIN-Programmen. Alle Kommentare in geschweiften Klammern werden entfernt und Gruppen von mehr als einem Leerzeichen werden auf eins reduziert. Dies ist eventuell aus Gründen der Speicherplatzersparnis im Flash-PROM der Steuerung angeraten. Allerdings ist die Einstellung nur geeignet, wenn die Programme nicht in der Steuerung angesehen oder geändert werden müssen.

Wert	Bedeutung
0	DIN-Programme ohne Veränderung ins Flash-PROM übernehmen (default).
1	Vor dem Speichern im Flash-PROM Kommentare entfernen und Programm reduzieren.

Hinweis!

Im Falle einer Rückübertragung von DIN-Programmen zum PC, bei der das Originalprogramm auf dem PC überschrieben wird, gehen auch dort die Kommentare verloren.

4.3.4 MK_NCPROG_NICHT_INS_EEPROM

Diese Maschinenkonstante verhindert bei Übertragung von DIN-Programmen über die Monitorschnittstelle die Übernahme der Programme in das interne Flash-PROM. Diese Option ist immer dann zu wählen, wenn das Einsatzgebiet der Steuerung verlangt, ständig neue DIN-Programme über die Monitorschnittstelle einzulesen, die nach dem Ausschalten der Maschinen nicht mehr benötigt werden. Dies ist normalerweise an Maschinen für Einzelauftragsfertigung üblich.

Wert	Bedeutung
0	DIN-Programme automatisch ins Flash-PROM übernehmen (default).
1	DIN-Programme nicht im Flash-PROM speichern.

Da die verwendeten Speicherbausteine nur eine begrenzte Anzahl Schreibzyklen (ca. 100000) vertragen, ist es in dem o. g. Fall sinnvoll, diese MK auf 1 zu setzen.

4.3.5 MK_METRISCH

Diese Maschinenkonstante wählt eines der beiden Maßsysteme für alle Weg- und Geschwindigkeitsangaben aus. Dies gilt für alle Eingabe- und Ausgabewerte für Wege und Geschwindigkeiten mit Ausnahme der Maschinenkonstanten selbst. Diese sind, unabhängig vom Inhalt dieser MK, immer im metrischen System einzutragen.

Wert	Bedeutung
0	Maßangaben laut MK_CONST_REL_INCH [inch] bzw. [inch/min]
1	Maßangaben laut MK_CONST_REL_MM [mm] bzw. [mm/min] (default)

4.3.6 MK_CONST_REL_MM

Diese Maschinenkonstante legt die Eingabeauflösung im metrischen System fest. Default wird 1, d. h. 1 mm / Einheit angenommen. Sollen alle Ein- und Ausgaben der Steuerung in µm-Einheiten erfolgen, so tragen Sie den Wert 0.001 ein.

4.3.7 MK_CONST_REL_INCH

Diese Maschinenkonstante legt die Eingabeauflösung im Inch-System fest. Default wird 25.4, d. h. 25.4 mm / Einheit angenommen. Wollen Sie Ein- und Ausgaben der Steuerung in 1/10tel Inch-Einheiten verstanden wissen, tragen Sie hier den Wert 2.54 ein.

4.3.8 MK_KONTURFEHLER

Diese Maschinenkonstante definiert die maximal zulässige (rechnerische) Abweichung [mm] von der programmierten Bahn bei einem nicht tangentialen Konturübergang. Sie beeinflusst damit zusammen mit MK_RADIUS_B_BEWERTUNG und MK_BAHNBESCHL, wie weit die Geschwindigkeit auf der Bahn an einer Ecke bei eingeschaltetem Look-Ahead (G60) reduziert werden muss.

Die Steuerung bestimmt dabei die maximale Geschwindigkeit an einer Ecke über folgenden Zusammenhang:

$$V_{\max} = \sqrt{R \cdot B}$$

V_{max} max.: Geschwindigkeit an der Ecke [mm/s]

R: Radius eines tangentialen Kreisbogens mit einem Abstand von MK_KONTURFEHLER zwischen dem programmierten Eckpunkt und dem Scheitelpunkt des Kreisbogens [mm]

B: Eingestelle Bahnbeschleunigung (MK_BAHNBESCHL, G201) [mm/s^2]

Zusätzlich wird die Geschwindigkeit durch den maximal zulässigen Ruck (Beschleunigungsanstieg pro Zeiteinheit) begrenzt:

$$V_{\max} = \sqrt[3]{R^2 \cdot B \cdot K/T}$$

T: Zeitkonstante für den zulässigen Beschleunigungsanstieg [s] MK_T_BAHNBESCHL

K: Wert von MK_RADIUS_B_BEWERTUNG

Generell gilt: Je kleiner der Wert in der MK, um so kleiner ist die resultierende Geschwindigkeit in Ecken und um so größer ist die Bahntreue. Diese Aussage gilt auch in ihrer Umkehrung.

4.3.9 MK_LAH_GRENZWINKEL

Diese Maschinenkonstante schränkt die Look-Ahead-Funktion (G60) auf einen definierten Winkelbereich ein. Dabei erfolgt ein automatisches Umschalten auf Genauhalt an nichttangentialen Satzübergängen, an denen der Betrag des Übergangswinkels größer als der Wert dieser MK ist.

Damit wird erreicht, dass solche Ecken in der Kontur auch bei eingeschaltetem Look-Ahead exakt angefahren werden, in dem die Geschwindigkeit an der Ecke zwangsweise auf 0 reduziert wird.

An den Satzübergängen, an denen der Übergangswinkel kleiner dem Wert dieser MK ist, wird die Geschwindigkeit nicht auf 0 reduziert, sondern nur soweit wie bei MK_KONTURFEHLER beschrieben.

Wert	Bedeutung
0	Jede noch so kleine Richtungsänderung führt zum Stillstand am Satzübergang.
15	Nur Richtungsänderungen von mehr als 15° führen zum Stillstand.
45	Nur Richtungsänderungen von mehr als 45° führen zum Stillstand (default).
180	Sonderfall: Keine Geschwindigkeitsreduzierung an Satzübergängen.

Erlaubte Werte für diese MK's liegen zwischen 0 und 180°. Bitte beachten Sie, dass der Sonderfall 180° zu Beschleunigungssprüngen an den Satzübergängen und dadurch zu falschen SW-Endschaltermeldungen bei der vorausschauenden SW-Endschalterfunktion (MK_SW_ENDS_MIT_RAMPE = 2) führen kann.

4.3.10 MK_RADIUS_B_BEWERTUNG

Diese Maschinenkonstante ist ein Faktor, mit dem die maximale Verfahrgeschwindigkeit beim Fahren auf Radien beeinflusst werden kann. Die Steuerung begrenzt die Geschwindigkeit beim Fahren von Kreisen in Abhängigkeit vom Kreisradius und der zulässigen Bahnbeschleunigungsänderung pro Zeiteinheit (Ruck) über folgenden Zusammenhang:

$$V_{\max} = \sqrt[3]{R^2 \cdot B \cdot K/T}$$

$V_{max:}$ max. Geschwindigkeit auf dem Kreisbogen [mm/s]

R: Radius des Kreisbogens [mm]

B: Eingestelle Bahnbeschleunigung (MK_BAHNBESCHL, G201) [mm/s^2]

T: Zeitkonstante für den zulässigen Beschleunigungsanstieg (MK_T_BAHNBESCHL) [s]

K: Wert von MK_RADIUS_B_BEWERTUNG

Diese MK stellt also einen Bewertungsfaktor für den zulässigen Ruck auf Kreisbögen dar. Mit ihr kann die Geschwindigkeitsbegrenzung für kleine Radien nach oben oder nach unten korrigiert werden.

4.3.11 MK_EPSILONMM

Diese Maschinenkonstante wird zur Toleranzbetrachtung [mm] bei der Angabe von translatorischen Positionen herangezogen. Sie wird zur Zeit nur bei der Kreisprogrammierung berücksichtigt und bestimmt dort, wie genau Kreisradius und Mittelpunkt programmiert werden müssen. Die Steuerung meldet bei der Interpretation eines Kreises immer dann einen Fehler, wenn der programmierte Kreisradius kleiner als diese MK ist oder der Betrag der Differenz zwischen dem Startradius und dem Zielradius größer ist als diese MK. Der Startradius ist dabei der Abstand zwischen dem programmierten Mittelpunkt und dem Startpunkt und der Endradius der Abstand zwischen dem Zielpunkt und dem programmierten Mittelpunkt des Kreises.

4.3.12 MK_EPSILONGRAD

Diese Maschinenkonstante wird zur Toleranzbetrachtung [grad] bei der Angabe von rotativen Positionen herangezogen. Sie hat nur Einfluss auf die Programmierung von richtungsabhängigen Modulo-360°-Achsen (s. MK_ACHSENART) und bestimmt dort, ob die programmierte Verfahrbewegung ausgeführt oder unterdrückt wird.

Beispiel:

Eine Rotationsachse soll auf +90° positioniert werden, wobei das Vorzeichen die Richtung angibt. Je nachdem ob die Achse vorher auf 89.5 oder auf 90.5 steht, müsste sie jetzt entweder um ein halbes Grad oder um 359.5 Grad verfahren werden. Hier greift MK_EPSILONGRAD ein, indem bei Verfahrwegen größer als 360°-MK_EPSILONGRAD im Absolutmaßsystem die Verfahrbewegung unterdrückt wird.

4.3.13 MK_OVERRIDEMAX

Diese Maschinenkonstante begrenzt die Overridevorgabe auf diesen Wert (Eingabe in 0,1 %).

4.3.14 MK_FEHLERRESTART

Diese Maschinenkonstante schaltet die Funktion ”Wiederanlauf nach Fehler” ein, mit der es möglich ist, nach einem Global-Leichten-Fehler (Fehlerklasse 3) im Programm an der unterbrochenen Stelle, oder an einer vorher im Programm definierten Stelle, wieder aufzusetzen. Nach einem solchen Fehler wird das Programm in einen Wartezustand versetzt, der mit Start (evtl. Start nach G10) für Wiederanlauf oder Stop für Abbruch verlassen werden muss.

Wert	Bedeutung
0	Kein Wiederanlauf möglich, Global-Leichter-Fehler führt zum Programmabbruch (default).
1	Wiederanlauf nach Fehler möglich.

Siehe auch Beschreibung zu G10 (📖 104).

4.3.15 MK_S0T0_VERSATZ_ERLAUBT

Diese Maschinenkonstante erlaubt das Verschieben der Koordinatensysteme S0 und T0, welche als Referenzkoordinatensysteme vorgesehen sind. Die Verschiebung dieser Koordinatensysteme ist normalerweise nicht erlaubt, damit sichergestellt ist, dass immer ein unverschobenes Koordinatensystem existiert.

In manchen Situationen kann es hilfreich sein, S0 oder T0 doch zu verschieben. Dann muss diese MK gesetzt werden.

Wert	Bedeutung
0	Koordinatenverschiebung von S0 oder T0 nicht erlaubt (default)
1	Koordinatenverschiebung von S0 oder T0 erlaubt.

Lesen Sie bitte auch die Beschreibungen zu den ”S-Funktionen” (📖 176) und ”T-Funktionen” (📖 177). Dort finden Sie wichtige Informationen zu den Koordinatensystemen.

4.3.16 MK_S_VERSATZSPERRE

Diese Maschinenkonstante schützt alle Werkstückkoordinatensysteme ab dem angegebenen Wert gegen ungewollte Verschiebung in DIN-Programmen. Der konfigurierte Wert wird in P613 im Parameterfeld der Steuerung eingeblendet.

Wert	Bedeutung
0	Verschiebung der Werkstückkoordinatensysteme S1 ... S99 ist erlaubt.
1 ... 99	Verschiebung der Werkstückkoordinatensysteme Sn ... S99 ist gesperrt, wobei n der bei MK_S_VERSATZSPERRE angegebene Wert ist.

4.3.17 MK_CANOPEN_BAUDRATE

Diese Maschinenkonstante legt die Bitraten für die beiden CAN-Busse der Steuerung fest. Diese MK hat zwei Parameter. Mit dem Ersten wird die Bitrate für den IO-CAN-Bus (CAN1) konfiguriert. Ein Wert von 0 bedeutet hier, dass keine CAN-Open-Module angeschlossen sind. Der zweite Parameter legt die Bitrate für den Drive-CAN-Bus (CAN2) fest.

Wert	Bedeutung
0	Keine CAN-Open-Module angeschlossen.
125	Bitrate 125 kBit/s
250	Bitrate 250 kBit/s
500	Bitrate 500 kBit/s
1000	Bitrate 1000 kBit/s

4.3.18 MK_DELTAT

Diese Maschinenkonstante stellt den internen Interpolationstakt [ms] der Steuerung ein. In jedem solchen Grobinterpolationstakt berechnet die Steuerung aus der programmierten Bahn neue Positionswerte für die beteiligten Achsen. Je kleiner dieser Takt ist, desto dichter liegen die berechneten

Positionen beieinander. Umso höher ist jedoch auch die Rechenbelastung für die Steuerung und die Busbelastung für Antriebsverstärker mit digitaler Schnittstelle. Die ETC kann mit einem minimalen Grobinterpolationstakt von 1 ms arbeiten.

Bei Verstärkern mit CAN-Schnittstelle ist die begrenzte Bandbreite der CAN-Bus-Übertragung von entscheidender Bedeutung. Die folgende Tabelle zeigt den minimalen Grobinterpolationstakt und die mittlere CAN2-Buslast in Abhängigkeit von der Anzahl der konfigurierten CAN-Achsen und der eingestellten Bitrate.

Bei der Berechnung der Buslast wurden nur die zyklischen Telegramme berücksichtigt, nicht eventuell zusätzlich anfallende asynchrone Telegramme von ETC-MMI oder SPS. Der angegebene Wert kann durch unterschiedliche Telegrammlängen bei verschiedenen Antriebsverstärkern leicht variieren.

Achsenanzahl	1000 kBit	500 kBit	250 kBit
1	1 ms / 29 %	2 ms / 29 %	3 ms / 38 %
2	1 ms / 52 %	2 ms / 52 %	4 ms / 52 %
3	2 ms / 38 %	3 ms / 50 %	5 ms / 60 %
4	2 ms / 49 %	3 ms / 66 %	6 ms / 66 %
5	2 ms / 61 %	4 ms / 61 %	7 ms / 70 %
6	2 ms / 73 %	4 ms / 73 %	8 ms / 73 %
7	3 ms / 56 %	5 ms / 67 %	9 ms / 75 %
8	3 ms / 64 %	5 ms / 77 %	10 ms / 77 %
9	3 ms / 72 %	6 ms / 72 %	11 ms / 78 %
10	3 ms / 79 %	6 ms / 79 %	12 ms / 79 %
11	4 ms / 65 %	7 ms / 75 %	13 ms / 80 %
12	4 ms / 71 %	7 ms / 81 %	14 ms / 82 %

Wenn sichergestellt ist, dass im Betriebszustand keine asynchronen Telegramme von MMI oder SPS übertragen werden, kann die zur Verfügung stehende Übertragungsbandbreite besser ausgenutzt werden, siehe folgende Tabelle (ohne Gewähr):

Achsenanzahl	1000 kBit	500 kBit	250 kBit
1	1 ms / 29 %	1 ms / 57 %	2 ms / 57 %
2	1 ms / 52 %	2 ms / 52 %	3 ms / 69 %
3	1 ms / 75 %	2 ms / 75 %	4 ms / 75 %
4	2 ms / 49 %	3 ms / 66 %	5 ms / 79 %
5	2 ms / 61 %	3 ms / 81 %	6 ms / 81 %
6	2 ms / 73 %	4 ms / 73 %	7 ms / 83 %
7	2 ms / 84 %	4 ms / 84 %	8 ms / 84 %
8	2 ms / 96 %	4 ms / 96 %	8 ms / 96 %
9	3 ms / 72 %	5 ms / 86 %	9 ms / 96 %
10	3 ms / 79 %	5 ms / 95 %	10 ms / 95 %
11	3 ms / 87 %	6 ms / 87 %	11 ms / 95 %
12	3 ms / 95 %	6 ms / 95 %	12 ms / 95 %

4.4 Speicherplatzreservierung

Die folgenden MK's beeinflussen das statische Speicherlayout innerhalb der Steuerung. Die eingestellten Werte sind erst nach einem Neustart der Steuerung wirksam, nachdem sie an der Steuerung geändert oder vom PC übertragen wurden.

4.4.1 MK_SPS_SPEICHERGROESSE

Diese Maschinenkonstante reserviert den Programmspeicher für die SPS im internen Arbeitsspeicher der Steuerung und legt damit gleichzeitig die maximale Größe des zu ladenden SPS-Programms in kByte fest. Die Steuerung reserviert das Doppelte des in der MK angegebenen Speichers, um die Funktion "Online-Change" der SPS-Programmieroberfläche zu unterstützen. Der so reservierte Speicher ist fest der SPS zugeordnet und steht für andere Aufgaben der Steuerung nicht mehr zur Verfügung.

Der Defaultwert von 128 kByte genügt für kleinere SPS-Programme. Bei größeren Programmen ist es notwendig, diesen Wert entsprechend zu vergrößern.

4.4.2 MK_SPS_DATENGROESSE

Diese Maschinenkonstante legt die Größe des Datenspeichers für die Variablen der SPS in kByte fest. Dieser Speicher wird fest für die SPS reserviert und steht in der Steuerung für andere Aufgaben nicht mehr zur Verfügung.

Der Defaultwert von 32 kByte sollte für die meisten SPS-Programme genügen.

4.4.3 MK_SPV_SPEICHERGROESSE

Diese Maschinenkonstante legt die Größe des internen Arbeitsspeichers im RAM zur Verwaltung von DIN-Programmen (auch Online-Programme) fest. Die Angabe versteht sich als Anzahl Speicherblöcke zu 1024 Byte (1 kByte). Ihr Wert kann bei Bedarf vergrößert oder verkleinert werden. Sinnvolle Werte beginnen ab 64 K. Der Maximalwert bei 4 MByte Speicherausbau beträgt 2048 kByte.

Hinweis!

Der Speicherbedarf der DIN-Programme im internen Arbeitsspeicher ist ca. 2 ... 4 mal so groß wie die Größe der entsprechenden ASCII-Datei. Dies liegt daran, weil die Programme zur schnelleren Ausführung beim Laden bereits vorinterpretiert und in einem Zwischencode im Arbeitsspeicher abgelegt werden.

4.4.4 MK_SPV_SYMBOLANZAHL

Diese Maschinenkonstante definiert die Größe der Symboltabelle, die für die symbolische Programmnummernverwaltung benötigt wird. Wenn bei dieser MK ein Wert größer 0 eingetragen wird, können bei der Programmierung der DIN-Programme symbolische Programmnamen verwendet werden. Dabei werden den Programmnamen dynamisch Programmnummern zwischen 32768 ... 65534 zugeordnet. Weitere Informationen zu diesem Thema finden Sie in Kapitel ”CNC-Programmierung”.

4.4.5 MK_HEADERANZAHL

Diese Maschinenkonstante bestimmt die Anzahl der ”Kopf-Einträge” zur Verwaltung der DIN-Programme im Arbeitsspeicher. Für jedes DIN-Programm im Arbeitsspeicher wird ein solcher Eintrag benötigt. Jeder Kopf-Eintrag belegt 32 Byte im internen Arbeitsspeicher (s. MK_SPV_SPEICHERGROESSE).

Mit dem Defaultwert von 100 können damit max. 100 DIN-Programme gleichzeitig im Arbeitsspeicher gehalten werden.

4.4.6 MK_KANALANZAHL

Diese Maschinenkonstante bestimmt die Anzahl der NC-Kanäle, in denen parallel DIN-Programme ausgeführt werden können.

Wert	Bedeutung
1	Es wird nur ein Kanal zur Ausführung von DIN-Programmen bereitgestellt (default).
2	Zwei Kanäle werden zur gleichzeitigen Ausführung von DIN-Programmen bereitgestellt.

Hinweis!

Für jeden Kanal werden zusätzliche Ressourcen an Rechenzeit und Arbeitsspeicher benötigt. Daher sollte diese MK nur vergrößert werden, wenn wirklich mehrere NC-Kanäle benötigt werden.

Bitte beachten Sie, dass ein Programm im zweiten Kanal nicht über die Starttaste gestartet werden kann, sondern nur aus einem Programm im ersten Kanal. Weitere Informationen zur Mehrkanalität finden Sie im Kapitel ”CNC-Programmierung” in den Beschreibungen zu G22 (📖 107) und G153 (📖 148).

4.4.7 MK_LAH_VORLAUFTIEFE

Diese Maschinenkonstante legt die maximale Anzahl der Aufträge im Vorlaufpuffer fest. Der Vorlaufpuffer ist ein Ringpuffer (FIFO) zwischen DIN-Interpreter und Grobinterpolator und sorgt für eine Entkopplung von Interpretation und Ausführung von DIN-Sätzen. Diese MK beeinflusst, um wie viel Sätze bei der Bearbeitung eines DIN-Programms vorausgeschaut werden kann.

Die MK enthält einen Parameter für jeden möglichen NC-Kanal.

Der minimale Wert dieser MK beträgt 8 Sätze. Der eingetragene Wert sollte eine 2er-Potenz sein (8, 16, 32, 64, 128, ...). Wenn nicht, wird er intern auf die nächst größere 2er-Potenz aufgerundet.

Der Vorlaufpuffer belegt Ressourcen im Arbeitsspeicher und zwar ca. 1500 Byte pro Satz. Falls nicht genug Speicher zum Anlegen des Vorlaufpuffers zur Verfügung steht, wird beim Neustart der Steuerung eine entsprechende Fehlermeldung generiert und der Vorlaufpuffer auf die Minimalgröße von 8 Sätzen reduziert.

Lesen Sie bitte auch das Kapitel "Satzvorverarbeitung und Zeitsynchronisation" in Kapitel "CNC-Programmierung" (📖 89).

4.4.8 MK_LAH_RUECKLAUFGRENZE

Diese Maschinenkonstante bestimmt die maximale Anzahl der Sätze, die ein Programm zurück interpoliert werden kann. Diese Sätze werden im Vorlaufpuffer gespeichert und reduzieren daher die Vorlauftiefe, siehe oben.

Die MK enthält einen Parameter für jeden möglichen NC-Kanal.

Der Wert in dieser MK darf nicht größer sein als MK_LAH_VORLAUFTIEFE-8. Sonst wird die Rücklaufgrenze auf diesen Wert begrenzt und ein Fehler gemeldet.

Hinweis!

Die Anzahl der rückwärts zu interpolierenden Sätze wird zusätzlich noch von anderen Faktoren begrenzt. Die Rückwärtsinterpolation geht generell nur soweit, wie Interpolationssätze vorhanden sind. Sie stoppt, sobald eine Funktion auftritt, die keine reine Interpreterfunktion ist und nicht in den vorhergehenden oder nachfolgenden Satz eingefügt werden kann, z. B. eine einzelne M-Funktion, vor der kein Verfahrsatz oder ein Verfahrsatz mit M- oder Q-Funktion steht (📖 172).

Bitte beachten Sie, dass während der Rückwärtsinterpolation standardmäßig M-Funktionen aber keine Q-Funktionen ausgegeben werden.

4.4.9 MK_PFELD_GROESSE

Diese Maschinenkonstante legt die Gesamtgröße des Parameterfeldes in der Steuerung fest und zwar inklusive der 1024 Systemparameter. Mit der MK wird die Anzahl der Parameter im Parameterfeld angegeben. Je Parameter werden 8 Byte Arbeitsspeicher belegt. Die Mindestgröße beträgt 2048 Parameter.

Weitere Informationen zum Parameterfeld finden Sie im Kapitel ”P-Feld” (📖 178).

4.5 Konfiguration der Achsen - Grundlegendes

Dies sind die wichtigsten Einstellungen, die vorgenommen werden müssen, da mit ihnen die Steuerung an die Mechanik angepasst wird.

Da die Steuerung für verschiedenste Achskombinationen eingesetzt werden kann, muss über die Maschinenkonstanten bekannt gegeben werden,

- ► welche Art Achsen an der konkreten Maschine vorhanden sind,
- ► unter welchem Kennbuchstaben die Achsen angesprochen werden sollen,
- ► auf welchem (Hardware) Achsrechnerkanal die entsprechende (Software) Achse angeschlossen ist.

In der Steuerung sind die Achsen von 0 ... N-1 durchnummeriert, wobei N die maximale Anzahl der konfigurierbaren Applikations-(=Software)-Achsen darstellt. Diese Anzahl ist steuerungsspezifisch und muss nicht mit der maximalen Anzahl der physikalischen (=Hardware) Achsen übereinstimmen. Am Ende dieses Kapitels finden Sie eine Tabelle, in der die maximale Achsanzahl der einzelnen Steuerungstypen aufgeführt ist.

Nachdem bekannt ist, welche physikalischen Achsen an der Maschine vorhanden sind, muss die Zuordnung der Applikationsachsen zu den physikalischen Achsen getroffen werden.

4.5.1 MK_CANDRIVES

MK_CANDRIVES definiert die Achsen, die am zweiten CAN-Bus angeschlossenen sind.

In jedem Parameter dieser MK ist eine gültige Applikationsachsnummer (0 ... N-1) oder der Wert -1 einzutragen, wobei N die maximale Anzahl der Applikationsachsen ist. Wenn zwei physikalische Achsen einer Applikationsachse zugewiesen sind, also eine Applikationsachsnummer zweimal in einer dieser MK auftaucht, spricht man von einer Synchronachse. Dies ist der einzige Fall, bei dem sich die Anzahl der konfigurierten physikalischen Achsen von der Anzahl der konfigurierten Applikationsachsen unterscheidet.

Nachdem bekannt ist, welche Applikationsachsen vorhanden sind, müssen diesen Achsen noch Buchstaben zugeordnet werden. Denn nur über die Achsbuchstaben können die Achsen im DIN-Programm programmiert werden.

4.5.2 MK_APPLACHSIDX

Diese MK hat 18 Parameter, einen für jeden der 18 möglichen Achsbuchstaben. Die Reihenfolge der den Parametern zugeordneten Buchstaben ist fest. Lediglich die Zuordnung der Applikationsachsnummern ist frei wählbar. Die Reihenfolge der Buchstaben lautet

X Y Z C U V W A B u v w a b c x y z

Jedem der Buchstaben kann genau eine Applikationsachse zugeordnet werden. Bei den nicht verwendeten Buchstaben sollten Sie eine -1 oder eine beliebige nicht konfigurierte Applikationsachsnummer eintragen.

Die Steuerung sucht nur die in MK_CANDRIVES konfigurierten Achsen und ordnet diesen den zuerst gefundenen Achsbuchstaben zu. Wenn Sie also eine Applikationsachsnummer mehrfach in dieser MK angeben, wird der erste Buchstabe verwendet und die anderen ignoriert.

Die Vergabe der Achsbuchstaben ist Ihnen vollkommen freigestellt. Jedoch sollten Sie die folgenden Regeln beachten, die das Verständnis Anderer für Ihre Maschine erheblich erleichtern.

Bezeichnen Sie die linearen Hauptachsen der Maschine mit X, Y, Z und die rotativen Hauptachsen mit A, B, C. Definieren Sie X, Y, Z so, dass Sie ein rechtsdrehendes Werkstückkoordinatensystem beschreiben und dass sich die Drehachsen A um die X-, B um die Y- und C um die Z-Achse drehen. Die Achsen U, V, W sollten für lineare Hilfsachsen verwendet werden, wobei sich U parallel zu X, V parallel zu Y und W parallel zu Z bewegen.

Mit MK_APPLACHSIDX legen Sie durch eine numerische Zuordnung die applikationsbezogene Reihenfolge der Achsbuchstaben fest. Diese Reihenfolge der Achsen gilt dann für alle achsbezogenen Maschinenkonstanten!

Beispiel

Sie haben die Aufgabe, eine Maschine zu konfigurieren, die eine X-, Z- und C-Achse hat, wobei die X-Achse eine Synchronachse sein soll.

Hierfür müssen Sie MK_CANDRIVES wie folgt besetzen:

MK_CANDRIVES	0,	/* X-Achse */
	0,	/* X'-Achse */
	1,	/* Z-Achse */
	2,	/* C-Achse */
	-1,	/* nicht belegt */
	-1,	
	-1,	
	-1;	
MK_APPLACHSIDX	0,	/* X */
	-1,	/* Y */
	1,	/* Z */
	2,	/* C */
	-1,	/* U */
	-1,	/* V */
	-1,	/* W */
	-1,	/* A */
	-1,	/* B */
	-1,	/* u */
	-1,	/* v */
	-1,	/* w */
	-1,	/* a */
	-1,	/* b */
	-1,	/* c */
	-1,	/* x */
	-1,	/* y */
	-1;	/* z */

Der Wert -1 gibt hierbei an, dass diese Achse nicht konfiguriert ist. Die X-Achse ist als erste, die Z-Achse als zweite und die C-Achse als dritte Applikationsachse bestimmt. D. h. bei allen nachfolgenden MK's dieser Rubrik wird der erste Parameter der Achse "X", der zweite Parameter "Z" und der dritte Parameter der "C"-Achse zugeordnet.

Die Anzahl der konfigurierbaren Achsen richtet sich nach der Ausbaustufe der Steuerung.

4.6 Konfiguration der Achsen - Zuordnung und Auswertung

4.6.1 MK_CANDRIVES

Diese Maschinenkonstante ordnet den Knotennummern 1 ... 12 am zweiten CAN-Bus Applikationsachsnummern zu und legt damit fest, welche Applikationsachsen konfiguriert sind. Für jede nicht belegte Knotennummer muss der Wert -1 eingetragen werden. Die Zuordnung der Achsbuchstaben zu den angegebenen Applikationsachsnummern erfolgt über MK_APPLACHSIDX.

Die MK hat maximal 12 Parameter für die 12 möglichen Knotennummern am Bus. Die Zuordnung der Knotennummern zu den Parametern ist fest von 1 ... 12 durchnummeriert.

Wird ein Index an zwei Stellen innerhalb von MK_CANDRIVES eingetragen, wird eine Zwangskopplung zwischen zwei Achsen erzeugt (Synchronachsen). Näheres hierzu ist dem Kapitel ”Synchronachsen” zu entnehmen (📖 217).

4.6.2 MK_APPLACHSIDX

Diese Maschinenkonstante ordnet den mit MK_HARDKONF, MK_ESABKONF und MK_CANDRIVES konfigurierten Applikationsachsnummern Buchstaben zu und definiert damit, welche Applikationsachsbuchstaben von der Steuerung akzeptiert werden.

Diese MK hat 18 Parameter, einen für jeden der 18 möglichen Achsbuchstaben. Die Reihenfolge der den Parametern zugeordneten Buchstaben ist fest. Lediglich die Zuordnung der Applikationsachsnummern ist frei wählbar. Die Reihenfolge der Buchstaben lautet

X Y Z C U V W A B u v w a b c x y z

Bitte beachten Sie, dass die Reihenfolge der Parameter aller nachfolgenden achsspezifischen MK den Applikationsachsnummern entspricht, d. h. 1. Parameter = Achsnummer 0, 2. Parameter = Achsnummer 1, n. Parameter = Achsnummer n-1, wobei die nicht konfigurierten Applikationsachsnummern mitgezählt werden.

Damit bestimmt diese MK, in welcher Reihenfolge die Parameter der achsspezifischen MK, bezogen auf die Achsbuchstaben, angegeben werden müssen.

Bitte beachten Sie auch die Ausführungen im vorherigen Kapitel!

4.6.3 MK_ACHSENART

Diese Maschinenkonstante spezifiziert verschiedene Achseigenschaften. Die MK ist Bit-codiert aufgebaut.

Bit	Wert	Bedeutung
0	0 1	Linearachse Rotationsachse
1	0 2	Endschalter beachten Endschalter ignorieren
2 ... 3	0 4 8 12	normale Achse Spindel Messachse Spindel + Messachse
4... 5	0 16 48	normale Achse Modulo-360°-Achse, das Vorzeichen gibt die Verfahrrichtung an Modulo-360°-Achse, die Winkellage wird auf dem kürzesten Weg angefahren [1)]
6	0 64	normale Achse Gantry-Achse, nur bei Synchronachsen wirksam
7	0 128	normale Achse Handrad

Um das gewünschte Bit zu setzen, müssen Sie den Wert aus der Tabelle zum Wert der MK dazu addieren.

Beispiel

Für eine Rotationsachse, die die Endschaltersignale nicht beachten soll und Modulo 360° programmiert wird, wobei das Vorzeichen die Verfahrrichtung angibt, müssten Sie den Wert 19 eintragen (Bit 0, 1 und 4).

Bitte beachten Sie, dass Handräder innerhalb der Steuerung durchnummeriert werden. Wenn Sie also mehrere Handräder deklarieren, wird das erste (das mit der kleinsten Applikationsachsnummer) als Handrad 0 bezeichnet, das nächste als Handrad 1, usw..

Eine Gantry-Achse stellt eine spezielle Form einer Synchronachse dar. Bei dieser besteht eine mechanische Kopplung zwischen Master- und Slave-Achse. Im Gegensatz zu einfachen Synchronachsen wird bei Gantry-Achsen mit analoger Schnittstelle die Referenzpunktfahrt synchron ausgeführt.

4.7 Konfiguration der Achsen - Auflösung

4.7.1 MK_IMPULSE

Diese Maschinenkonstante bestimmt die Anzahl der Impulse, die der Istwertzähler des Achsinterface bei rotatorischen Motoren pro Umdrehung an der Motorwelle und bei Linearmotoren pro Millimeter bekommt. Einzutragen ist die Impulsanzahl inklusive Impulsvervierfachung; z. B. beim ECS-Kompaktservo 65536 Imp. / Umdrehung.

Durch Angabe eines negativen Wertes wird die Definitionsrichtung der Achse umgedreht, d. h. sie fährt in die andere Richtung. Der gleiche Effekt tritt auf, wenn Sie stattdessen bei der Angabe von MK_WEG das Vorzeichen drehen.

Das gilt auch für die Referenzpunktfahrt (zu berücksichtigen bei MK_REF_RICHTUNG_UND_FOLGE)!

4.7.2 MK_WEG

Diese Maschinenkonstante legt die Wegstrecke fest, bei der der Achsrechner die unter MK_IMPULSE eingetragene Anzahl Inkremente zählt, d. h. bei rotatorischen Motoren der Weg pro Umdrehung an der Motorwelle und bei Linearmotoren 1 mm. Die Einheit bei Linearachsen ist Millimeter, bei Rotationsachsen Grad.

Beispiel

MCS06-Servomotor mit ECS-Kompaktservo (65536 Impulse): Der Motor geht über ein Getriebe (4:1) an eine Spindel, die einen Schlitten pro Umdrehung um 3.4 mm rückwärts bewegt. D. h. bei einer positiven Wegvorgabe bewegt sich die Mechanik in negativer Richtung.

```
MK_IMPULSE  65536,...;
MK_WEG     -0.85,...;   /* 3.4 / 4 */
```

4.7.3 MK_MASSSTAB

Diese Maschinenkonstante setzt einen Multiplikationsfaktor für achsspezifische Wegvorgaben und Positionsanzeigen fest. Dies ist hilfreich, wenn bei runden Werkstücken Durchmessermaße programmiert werden, die Vorschubgeschwindigkeit aber radiusbezogen sein soll.

Ein Maßstabfaktor für X von 0.5 bedeutet, dass eine Wegvorgabe von 10 mm eine Verfahrbewegung von X um 5 mm zur Folge hat. In der Anzeige erscheinen aber die 10 mm.

4.8 Konfiguration der Achsen - Arbeitsfeld

4.8.1 MK_GRUNDOFFSET

Diese Maschinenkonstante ist der Versatz des mechanischen Nullpunktes der Maschine zum Nullpunkt des Wegmesssystems. Die Einheit ist Millimeter für Linearachsen und Grad für Rotationsachsen.

4.8.2 MK_SW_ENDS_MINUS, MK_SW_ENDS_PLUS

Diese Maschinenkonstanten definieren die positive und negative Verfahrbereichsgrenze der Mechanik. Die Einheit ist Millimeter für Linearachsen und Grad für Rotationsachsen. Die Überwachung muss mit G26 eingeschaltet werden. Siehe auch MK_VOREINSTELLUNG.

Die Werte beziehen sich auf den Maschinennullpunkt (Referenzpunkt + Grundoffset) und nicht auf den Nullpunkt des Wegmesssystems.

4.8.3 MK_SW_ENDS_MIT_RAMPE

Diese Maschinenkonstante zeigt an, ob bei Überschreiten der Verfahrbereichsgrenzen kontrolliert gebremst oder abrupt stehen geblieben werden soll.

Wert	Bedeutung
0	Beim Erreichen der SW-Endschalter schlagartig anhalten.
1	Auf der programmierten Bahn mit der aktuellen Bremsrampe abbremsen. Dabei muss die Verfahrbereichsgrenze so gewählt sein, dass die Achse mit den eingestellten Beschleunigungswerten noch vor dem mechanischen Anschlag zum Stehen kommen kann.
2	Vorausschauende Software-Endschalter. Dabei wird anhand der aktuellen Achsgeschwindigkeiten und der konfigurierten max. Achsbremsrampe berechnet, ob der entsprechende Software-Endschalter erreicht wird. Ist dies der Fall, wird die betreffende Achse so abgebremst, dass sie an der eingestellten Softwaregrenze zum Stehen kommt. Alle anderen an der Bahn beteiligten Achsen werden wie bei 1 angehalten. Achtung: Ein korrektes Arbeiten dieser Funktion kann nur gewährleistet werden, wenn in MK_LAH_GRENZWINKEL ein Wert kleiner 180° eingetragen ist.

4.8.4 MK_SCHLEPPGENAUHALT

Diese Maschinenkonstante legt den Grenzwert der Regelabweichung der einzelnen Achsen für Genauhalt mit Schleppabstandsüberwachung in [mm] bzw. [Grad] fest. Diese Funktion muss mit G60 aktiviert werden (📖 116).

4.8.5 MK_GENAUHALTZEIT

Diese Maschinenkonstante legt die Zeit in Sekunden fest, innerhalb der der Betrag des Schleppabstands einer Achse den bei MK_SCHLEPPGENAUHALT konfigurierten Wert ständig unterschritten haben muss, bevor die nächste Funktion ausgeführt werden kann. Diese MK ist nur wirksam, wenn die Funktion ”Genauhalt mit Schleppabstandsüberwachung” eingeschaltet ist. Der Defaultwert ist 0.

4.9 Konfiguration der Achsen - Reglereinstellung

4.9.1 MK_T2

Diese Maschinenkonstante ist die Filterzeitkonstante für den Feininterpolationsfilter in Sekunden. Diese MK beeinflusst nicht direkt die Lageregelung, sondern die Feininterpolation, mit deren Hilfe neue Lagesollwerte im Feininterpolationsraster (MK_FIT_PRO_GIT) berechnet werden. Der Feininterpolationsfilter ist ein Filter 2. Ordnung, mit dem die berechneten Lagesollwerte geglättet werden können, um ein sanfteres Fahrverhalten bei bestimmten Achsen zu erzielen. Diese MK wirkt auch auf Achsen mit digitaler Schnittstelle.

Der Feininterpolationsfilter führt zu einer zeitlichen Verzögerung bei der Ausgabe der Lagesollwerte, die ca. 2mal so groß ist wie die Filterzeitkonstante. Daher sollte bei Achsen, die gemeinsam an einer Bahninterpolation beteiligt sind, grundsätzlich die gleiche Filterzeitkonstante eingetragen werden.

Diese MK ist defaultmäßig auf 0 gesetzt und sollte nur vergrößert werden, wenn eine Achse zum Schwingen bei Unstetigkeiten auf der Bahn neigt, z. B. an Übergängen von einem Kreisradius zu einem anderen Kreisradius. Dies ist meistens bei Achsen mit einem großen Umkehrspiel und bei Zahnstangen der Fall. Sinnvolle Werte liegen zwischen 0 und 0.2 s. Je größer der eingestellte Wert ist, desto größer ist die durch den Filter hervorgerufene Bahnabweichung.

4.10 Konfiguration der Achsen - Referenzierung

4.10.1 MK_REF_RICHTUNG_UND_FOLGE

Diese Maschinenkonstante bestimmt die Richtung und die Reihenfolge, in der die Achsen eine Referenzpunktfahrt ausführen. Die Richtung, in der die Achse die erste Verfahrbewegung ausführt, wird durch das Vorzeichen des eingetragenen Wertes bestimmt. Die Reihenfolge, in der die Achsen beim gemeinsamen Referenzieren die Refererenzpunktfahrt ausführen, wird durch den Betrag des Wertes festgelegt.

Wenn eine Achse nicht an der automatischen Referenzpunktfahrt beteiligt werden soll, muss für diese Achse 0 oder -128 eingetragen werden. Wobei -128 eine Richtungsumkehr bei der manuellen Referenzpunktfahrt der Achse bedeutet.

Beispiel

Die Maschine hat eine X-, Y-, Z- und eine C-Achse, wobei sichergestellt sein muss, dass die Z-Achse aus dem Arbeitsraum gefahren ist, bevor X und Y bewegt werden (MK_APPLACHSIDX mit Defaultwerten). Die Y-Achse soll in negativer Richtung den Nocken suchen. Die C-Achse soll nicht an der automatischen Referenzpunktfahrt beteiligt werden.

MK_REF_RICHTUNG_UND_FOLGE 2, -2, 1, 0;

4.11 Konfiguration der Achsen - Geschwindigkeit und Beschleunigung

Es gibt verschiedene MK's, um die maximalen Geschwindigkeiten und Rampen für die einzelnen Achsen und für die resultierende Bahn zu beeinflussen. Diese stellen die Grenzwerte dar, die von der Steuerung nicht überschritten werden dürfen. Die achsspezifischen MK's wirken im Bahn-, Strecken- und Handbetrieb. Die bahnspezifischen MK wirken nur im Bahnbetrieb begrenzend.

4.11.1 MK_MODVMAX

Diese Maschinenkonstante ist die maximale Geschwindigkeit für Handbetrieb (Verfahrtasten) der einzelnen Achsen. Der angegebene Wert wird auf MK_VMAX begrenzt.

4.11.2 MK_VMAX

Diese Maschinenkonstante ist die maximal zulässige Geschwindigkeit der einzelnen Achsen in [m/min] bzw. [1/min]. Diese entspricht der Eilgangsgeschwindigkeit der Achsen im Streckenbetrieb. Auch die Verfahrgeschwindigkeit im Bahnbetrieb wird, wenn möglich, von der Steuerung so begrenzt, dass die Maximalgeschwindigkeit der einzelnen Achsen nicht überschritten wird.

Hinweis!

Bei einigen Transformationen wie Grund-/Konturdrehung und 6-Achsen-Transformation kann die Steuerung die Bahngeschwindigkeit nicht begrenzen, da die Transformation, logisch gesehen, erst nach der Bahninterpolation erfolgt und die Geschwindigkeit der Achsen nicht vorher bestimmbar ist.

4.11.3 MK_BESCHL, MK_BREMS

Diese Maschinenkonstanten legen die maximalen Beschleunigungs- und Bremsrampen der einzelnen Achsen in [m/s^2] bzw. [$1/s^2$] fest. Im Bahnbetrieb werden die Bahnrampen entsprechend der Beteiligung der einzelnen Achsen anteilig begrenzt.

4.11.4 MK_T_BESCHL

Diese Maschinenkonstante ist die Zeit in Millisekunden, in der die unter MK_BESCHL und MK_BREMS konfigurierten maximalen Rampen im Handbetrieb und im Streckenbetrieb erreicht werden sollen. Diese MK dient zum Einstellen von $\sin^2$-ähnlichen Beschleunigungs- und Bremsrampen der einzelnen Achsen.

4.11.5 MK_VBAHNMAX

Diese Maschinenkonstante legt die maximale Geschwindigkeit [m/min] im Bahnbetrieb fest. Diese kann größer sein als die maximale Geschwindigkeit der einzelnen Achsen, wenn die resultierende Geschwindigkeit der beteiligten Achsen kleiner oder gleich MK_VMAX ist.

4.11.6 MK_BAHNBESCHL, MK_BAHNBREMS

Diese Maschinenkonstanten legen die maximal zulässigen Beschleunigungs- und Bremsrampen [m/s^2] im Bahnbetrieb fest. Die Bahnrampen werden zusätzlich in Abhängigkeit der beteiligten Achsen so begrenzt, dass die resultierenden Rampen der einzelnen Achsen die Werte von MK_BESCHL und MK_BREMS nicht überschreiten.

4.11.7 MK_T_BAHNBESCHL

Diese Maschinenkonstante ist die Zeit in Millisekunden, in der die unter MK_BAHNBESCHL und MK_BAHNBREMS angegebenen Rampen erreicht werden sollen. Diese MK dient zum Einstellen von sin^2-ähnlichen Beschleunigungs- und Bremsrampen auf der Bahn.

4.12 Konfiguration der Achsen - Achskorrektur

4.12.1 MK_SPINDELUMKEHRSPIEL

Diese Maschinenkonstante gibt in Millimeter bzw. Grad an, um welchen Wert die Sollposition bei Fahrrichtungsumkehr korrigiert werden soll. Default ist 0. Sobald ein Wert eingetragen ist, ist die Spindelumkehrkompensation eingeschaltet.

Korrektur eines Spindelsteigungsfehlers

Die Korrektur eines Spindelsteigungsfehlers oder Linearitätsfehlers des Wegmesssystems kann mit Hilfe einer Tabelle, die im Flash-PROM gespeichert oder vom ETC-MMI nach dem Hochfahren der Steuerung übertragen wird, erfolgen. Die Korrektur ist automatisch eingeschaltet, wenn die Tabelle übertragen wurde.

Die Korrektur erfolgt für die in der Datei angegebenen Achsen innerhalb der definierten Istpositionen. Die Positionen sind auf den Referenzpunkt bezogen. Bei einem sich periodisch wiederholenden Fehler besteht die Möglichkeit, über die Angabe eines ”Modulo”-Wertes den relativen Bereich innerhalb des vorgegebenen Korrekturbereichs zu definieren. Am Ende dieses relativen Bereichs werden wieder die Werte vom Anfang der Tabelle verarbeitet.

Für die Korrekturwerte steht jeweils ein Byte zur Verfügung, d. h. der Korrekturwert kann -128 ... 127 Inkremente pro Positionswert betragen. Der Korrekturwert wird bei der Achsausgabe auf die errechnete Position addiert. Zwischen zwei Stützstellen erfolgt eine lineare Interpolation. Für Positionen, die außerhalb des definierten Bereichs liegen, erfolgt keine Korrektur. Die Korrekturtabelle kann maximal 4096 Einträge pro Achse enthalten.

Aufbau der Korrekturdatei

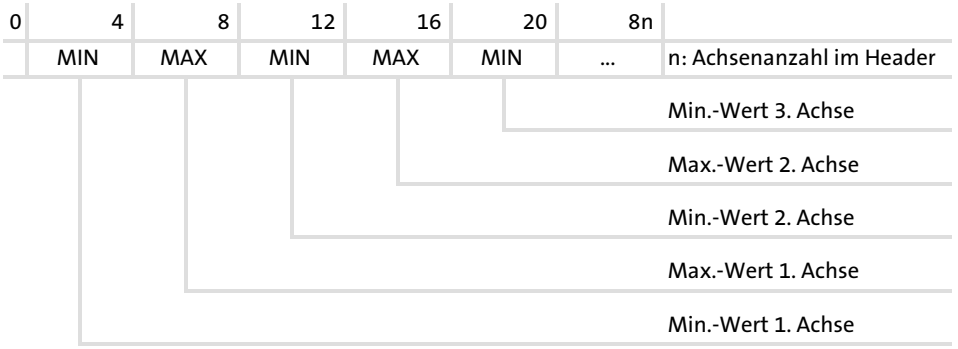
Die Datei ist als Binärdatei aufgebaut. Sie muss im Flash-PROM unter dem Namen ”ACHSEN.KOR” gespeichert sein (Laden mit Hilfsprogramm ”download”). Die Datei besteht aus einem 128 Byte langen Header, 2 Definitionsblöcken und bis zu 512 Datenblöcken zu je 8 Byte pro Achse. Die Daten werden im Binärformat abgelegt.

128	8n	8n	8n	8n
Header	Def 1: MIN/ MAX	Def 2: MOD	Daten (1)	Daten (512)

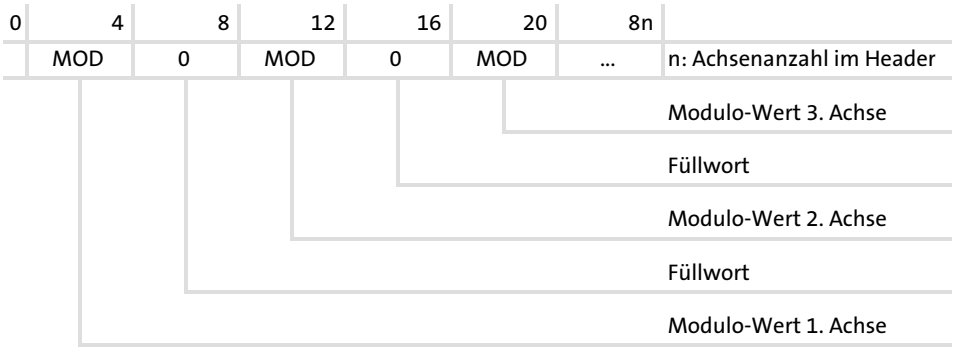
Header: Der Header enthält die nullterminierte Liste der Achsbuchstaben der beteiligten Achsen. Die Reihenfolge ist beliebig. Die Daten der nachfolgenden Blöcke werden in dieser Reihenfolge interpretiert. Der verbleibende Rest ist mit binären Nullen aufzufüllen.

0	1	2	3	n		127	128
	X	Y	Z	...	0	...	0

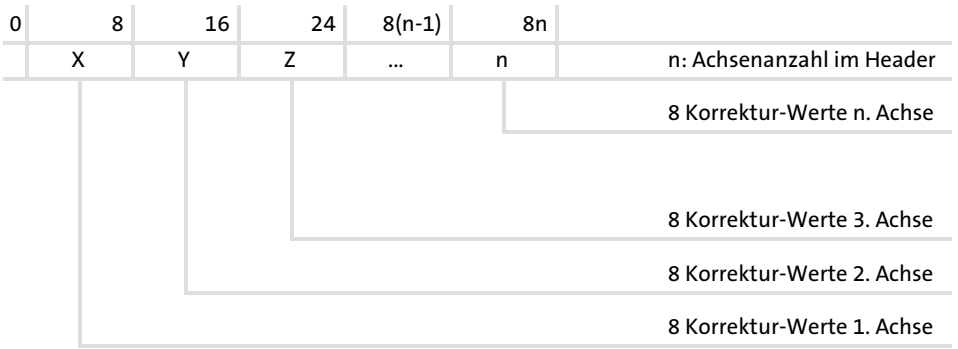
Def 1: Der erste Definitionsblock enthält pro Achse im Header je einen 32-Bit-Integer für Minimal- und Maximalwert des Istpositionsbereichs, in dem die Korrekturtabelle gültig ist. Die Angabe erfolgt in Inkrementen bezogen auf den Referenzpunkt.



Def 2: Der zweite Definitionsblock enthält pro Achse einen 32-Bit-Integer für den Modulowert. Der Modulowert definiert den relativen Istpositionsbereich, den die Korrekturtabelle abdeckt. Dieser Bereich darf sich zwischen Min- und Maxwert mehrfach wiederholen. Die Angabe erfolgt in Inkrementen. Die restlichen 4 Byte pro Achse sind nicht belegt und müssen mit 0 initialisiert werden.



Daten: Jeder Datenblock umfasst jeweils 8 Byte Daten pro Achse und enthält die eigentlichen Korrekturdaten. Jeder Korrekturwert wird durch ein Byte dargestellt. Somit sind in einem Datenblock 8 Korrekturwerte pro Achse enthalten. Jeder Korrekturwert hat einen Wertebereich von -128 ... +127 Inkrementen.



4.13 Konfiguration der Achsen - Handräder

Die Steuerung erlaubt den Anschluss von elektronischen Handrädern mit CAN-Interface. Dazu ist es notwendig, das Handrad in den Maschinenkonstanten zu konfigurieren. Die Konfiguration erfolgt analog einer normalen Achse.

Die konfigurierten Handräder werden in der Steuerung von 0 ... N-1 durchnummeriert, wobei N der maximalen Anzahl der konfigurierbaren Handräder entspricht. Die Vergabe eines Achsbuchstabens ist daher optional und wird im Allgemeinen nicht benötigt.

Folgende Maschinenkonstanten müssen gesetzt werden:

4.13.1 MK_CANDRIVES

Diese Maschinenkonstante ordnet den Knotennummern 1 ... 12 am zweiten CAN-Bus Applikationsachsnummern zu und legt damit fest, welche Applikationsachsen konfiguriert sind. Für jede nicht belegte Knotennummer muss der Wert -1 eingetragen werden. Die Zuordnung der Achsbuchstaben zu den angegebenen Applikationsachsnummern erfolgt über MK_APPLACHSIDX.

4.13.2 MK_APPLACHSIDX

Vergabe eines freien Achsbuchstabens für das konfigurierte Handrad. Die Zuordnung eines Buchstabens ist optional und wird nur dann benötigt, wenn das Handrad oder der konfigurierte Applikationsachskanal über andere G-Funktionen als G187 angesprochen werden soll, z. B. Referenzpunktfahrt mit G74.

4.13.3 MK_ACHSENART

Bei der mit MK_CANDRIVES zugeordneten Applikationsachse muss Bit 7 gesetzt bzw. der Wert 128 eingetragen werden.

4.13.4 MK_IMPULSE

Anzahl der Impulse, die das Handrad pro einer definierten Wegstrecke liefert.

4.13.5 MK_WEG

Weg, den eine zugeordnete Achse pro Anzahl Impulse (MK_IMPULSE) bei einem Bewertungsfaktor von 1.0 zurücklegen soll. Die Richtung kann mit dem Vorzeichen gedreht werden.

4.13.6 MK_HANDRADZUORDNUNG

Diese Maschinenkonstante ist eine achsspezifische MK, die den einzelnen Achsen ein konfiguriertes Handrad zuordnet. Diese MK hat einen Parameter für jede Applikationsachse, in dem die Nummer des zugeordneten Handrades eingetragen wird. Sie muss nur geändert werden, wenn mehr als ein Handrad konfiguriert ist. Die Zuordnung kann nur über diese MK geändert werden.

4.13.7 MK_HANDRADFAKTOR

Diese Maschinenkonstante ist ein zusätzlicher achsspezifischer Bewertungsfaktor für die Handradfunktion. Dieser wird verwendet, um unterschiedliche Bewertungsfaktoren für die einzelnen Achsen zu erzielen, z. B. dann, wenn an einem Handrad Linear- und Rotationsachsen betrieben werden.

4.13.8 MK_HANDRADFILTER

Diese Maschinenkonstante ist eine Filterzeitkonstante [ms] für die Handradfunktion. Mit ihr kann die Geschwindigkeitsvorgabe des Handrades geglättet werden, um die Verfahrbewegung über das Handrad weicher zu machen. Dies ist normalerweise nur bei hohen Handradbewertungsfaktoren und geringer Handradauflösung notwendig, z. B. Bewertungsfaktor 100 und ein Inkrement je Raster am Handrad.

Wert	Bedeutung
0	Handradfilter ausgeschaltet (default)
100	100 ms Filterzeitkonstante, entspricht einer Grenzfrequenz von 10 Hz
250	250 ms Filterzeitkonstante, entspricht einer Grenzfrequenz von 4Hz

Der Wert kann beliebig zwischen 0 und 500 ms eingestellt werden.

Der Handradfilter bewirkt eine zeitliche Verzögerung der Positionsvorgaben vom Handrad an die angewählte Achse um den eingestellten Wert. Deshalb sollte die Zeitkonstante nicht zu groß eingestellt werden, sonst werden die Reaktionszeiten unnötig verlängert.

4.14 Konfiguration der Achsen - Synchronachsen

Eine Synchronachse wird in der Steuerung konfiguriert, in dem die zugehörige Achsnummer an zwei Stellen innerhalb von MK_CANDRIVES eingetragen wird. Damit wird eine Zwangskopplung zwischen den beiden physikalischen Achsen erzeugt.

Der Achskanal mit dem niedrigeren Index gilt automatisch als Führungsachse, der nachfolgende Achskanal, dem die gleiche Achskennung zugeordnet wurde, ist die Folgeachse. Alle Vorgaben für die Führungsachse werden synchron von der Folgeachse ausgeführt.

Ist eine Zwangskopplung konfiguriert, müssen zusätzlich die folgenden Maschinenkonstanten bestimmt werden.

4.14.1 MK_ACHSENART

In dieser MK muss Bit 6 gesetzt werden, wenn die Synchronachse eine Gantry-Achse ist. Bei einer Gantry-Achse besteht eine mechanische Kopplung zwischen Führungsachse und Nachführachse. Dies bewirkt bei Achsen mit analoger Schnittstelle, dass bereits die Referenzpunktfahrt synchronisiert ausgeführt wird. Bei Achsen mit digitaler Schnittstelle wird die Referenzpunktfahrt lediglich synchron gestartet.

4.14.2 MK_SYNCHRONABWEICHUNG

Diese Maschinenkonstante gibt die maximal zulässige Abweichung zwischen Führungs- und Folgeachsenposition für die Synchronfehlerüberwachung an. Die Einheit ist je nach Achsenart Millimeter bzw. Grad. Die Maschinenkonstante ist nur von Belang, wenn eine Zwangskopplung konfiguriert wurde.

4.14.3 MK_SYNCHRONOFFSET

Diese Maschinenkonstante definiert den Abstand zwischen den Referenzpunkten von Führungsachse und Folgeachse in [mm] bzw. [Grad]. Dieser sollte im Idealfall 0 sein, damit beide Achsen nach dem Erreichen des Referenzpunktes bereits synchron sind. Die Folgeachse fährt nach dem Erreichen des Referenzpunktes um den hier eingetragenen Wert weiter, während die Führungsachse am Referenzpunkt stehen bleibt.

4.15 Technologiespezifische Einstellungen

4.15.1 MK_MFKT_UPR_TABELLE

Diese Maschinenkonstante ist eine Liste von bis zu 16 M-Funktionsnummern, die bei Bearbeitung innerhalb eines DIN-Programms eine Verzweigung in ein Unterprogramm bewirken. Dabei folgende Fälle unterschieden:

Wert	Bedeutung
< 500	Wenn die konfigurierte M-Funktion im Programm auftaucht, wird nach ihr ein Unterprogrammaufruf mit der Programmnummer 9000 + M-Funktionsnummer eingefügt. Die programmierte M-Funktion wird ausgegeben bevor der erste ausführbare Satz des Unterprogramms ausgeführt wird.
500 600 700 800 900	Jeder dieser Werte steht gleichzeitig stellvertretend für die 99 folgenden M-Funktionsnummern. Immer wenn eine dieser 100 M-Funktionen im Programm auftaucht, wird die M-Funktion durch einen Unterprogrammaufruf mit der Programmnummer 9000 + Basis-M-Funktionsnummer ersetzt. Aus M621 wird also G22 L9600. Die programmierte M-Funktion wird nicht ausgegeben, daher fällt auch keine Verarbeitungszeit für die M-Funktion in der SPS an.

Die M-Funktion, die zum Unterprogrammaufruf führte, wird im Parameterfeld in P562 für NC-Kanal 0 und in P594 für Kanal N abgelegt. Sie kann im Unterprogramm ausgewertet werden.

Weitere Informationen finden Sie im Kapitel ”M-Funktionen” (📖 173).

4.15.2 MK_TECHNOLOGIEDATEN1 ... MK_TECHNOLOGIEDATEN4

Diese Maschinenkonstanten enthalten jeweils 10 Einträge, die eins zu eins im Parameterfeld von P760 ... P799 abgelegt werden und dort beliebig verwendet werden können. Die Belegung des Parameterfeldes finden Sie im Kapitel ”P-Feld” (📖 178).

4.15.3 MK_MASCH_POLAR_KART

Diese Maschinenkonstante schaltet von einen kartesischen auf ein polares Maschinenkoordinatensystem um. Im Polarkoordinatensystem werden die Positionen in der Maschinenebene über eine Winkel- und eine Radiusachse bestimmt. Wobei die Steuerung die Transformation der interpolierten kartesischen Koordinaten ins Polarkoordinatensystem übernimmt.

Programmierung und Interpolation erfolgen dabei weiterhin in kartesischen Koordinaten, und zwar sowohl beim Handverfahren als auch im Programmbetrieb. Die Zuordnung der kartesischen und polaren Achsen zu den Koordinatensystemen erfolgt über MK_KARTESISCH_ACHSNR und MK_POLAR_ACHSNR.

Wert	Bedeutung
0	kartesisches Maschinenkoordinatensystem (default)
1	polares Maschinenkoordinatensystem

Hinweis!

Die Geschwindigkeitsbegrenzung innerhalb des Interpolators erfolgt lediglich im kartesischen Koordinatensystem. Die maximalen Geschwindigkeiten der polaren Radius- und Winkelachse nach der Transformation können nicht überwacht werden. Im Extremfall, z. B. Fahren durch den polaren Nullpunkt, kann es daher zum Schleppfehler kommen.

4.15.4 MK_KARTESISCH_ACHSNR

Diese Maschinenkonstante legt die Nummern der kartesischen Achsen bei einem polaren Maschinenkoordinatensystem fest. Aus den Positionen dieser virtuellen Achsen werden später die Positionen der polaren Radius- und Winkelachse berechnet. Die Nummern der Achsen sollten dieselben wie die der polaren Achsen sein, damit nicht mehr Achsen konfiguriert werden müssen, als physikalisch vorhanden sind.

Diese MK hat die folgenden drei Einträge:

Index	Default	Bedeutung
0	0	Nummer der Cosinus-Achse (X)
1	1	Nummer der Sinus-Achse (Y)
2	-1	Nummer der optionalen tangentialen Nachführachse oder -1 (C)

Die ersten beiden eingetragenen Achsen müssen in MK_ACHSENART als Linearachsen konfiguriert sein. Alle Einheiten abhängigen MK außer MK_WEG sind für diese Achsen in [mm] anzugeben. Die dritte Achse muss als Rotationsachse konfiguriert werden.

4.15.5 MK_POLAR_ACHSNR

Diese Maschinenkonstante legt die Nummern des Radius und der Winkelachse im polaren Maschinenkoordinatensystem fest. Dies sind die physikalischen Achsen für Radius und Winkel an der Maschine.

Diese MK hat die folgenden 2 Einträge:

Index	Default	Bedeutung
0	0	Nummer der Radius-Achse
1	1	Nummer der Winkel-Achse

Da dies dieselben Achsnummern wie die ersten beiden Achsen bei MK_KARTESISCH_ACHSNR sein sollten, ist auch die restliche Konfiguration gleich. Lediglich bei MK_WEG muss bei der Winkelachse die Einheit Grad verwendet werden, da es sich dabei reell um eine Rotationsachse handelt.

4.15.6 MK_WLK_C_GRENZWINKEL

Diese Maschinenkonstante wird von verschiedenen Korrekturmodulen (z. B. ”SCHNEIDEN”), für die tangentiale Nachführung der C-Achse verwendet und definiert, bis zu welchem Relativwinkel eine schlagartige Zustellung dieser Achse erfolgen darf.

Bei Winkeln größer als diese MK, wird der Vorschub auf der Bahn gestoppt, die C-Achse einzeln mit G0 zugestellt und anschließend die Bahn weiterinterpoliert. Je nach Korrekturmodul wird vor der G0 noch eine M15 und nach der G0 eine M14 eingefügt, um die Zustellung bei angehobenem Werkzeug durchzuführen.

Bei kleineren Winkeln erfolgt die Zustellung der C-Achse auf den neuen Konturwinkel am Satzübergang in einem Schritt, ohne die Bahninterpolation zu beeinflussen.

Wert	Bedeutung
45	schlagartige Zustellung bis 45° (default)
15	schlagartige Zustellung bis 15°
90	schlagartige Zustellung bis 90°

Jeder beliebige Winkel zwischen 0 und 180° kann eingestellt werden.

Hinweis!

Welches Korrekturmodul bei G41/G42 eingeschaltet wird, ist technologiespezifisch und abhängig vom Wert in MK_KUNDE (📖 191).

4.15.7 MK_WLK_C_OFFSET

Diese Maschinenkonstante wird im Korrekturmodul ”Schneiden” verwendet, um eine technologiespezifische Schrägstellung des Werkezugs auszugleichen. Die MK gibt den Offset in Grad an, um den die C-Achse gegenüber der berechneten Bahnorientierung verschoben sein soll. Der Defaultwert von 0 bewirkt keine Verschiebung.

4.15.8 MK_WLK_VERWEILZEIT

Diese Maschinenkonstante dient zum automatischen Einfügen von Verweilzeiten im Korrekturmodul "SCHNEIDEN" an allen Stellen, an denen das Schneidwerkzeug aus- oder eingestochen werden muss. Die MK hat zwei Parameter. Der erste Eintrag definiert die Verweilzeit nach dem Ausstechen (M15/M16) und der zweite die Verweilzeit nach dem Einstechen. Beide werden in Sekunden mit Nachkommastellen eingetragen.

4.15.9 MK_X_WINKEL

Diese Maschinenkonstante definiert ein schrägwinkliges Koordinatensystem in der Z-X-Ebene, wie es z. B. bei Rundschleifmaschinen mit schräggestellter Schleifscheibe benötigt wird. Mit dieser MK wird angegeben, um wie weit der Winkel der X-Achse von der Senkrechten abweicht.

In der Anzeige und im Parameterfeld wird immer die Werkzeugspitze relativ zum Werkzeug in rechtwinkligen Koordinaten angezeigt. Beim Fahren der X-Achse wird zwischen Bahn- (G31) und Streckenbetrieb (G30) unterschieden. Während sich im Bahnbetrieb in der Anzeige nur X bewegt, fahren an der Maschine X und Z. Die Werkzeugspitze bleibt in Z-Richtung relativ zum Werkzeug stehen.

Im Streckenbetrieb ist das Verhalten genau umgekehrt. Während an der Maschine nur die X-Achse fährt, bewegt sich in der Anzeige X und Z. Die Werkzeugspitze wandert in Z-Richtung relativ zum Werkstück weg.

Hinweis!

Beim Fahren der X-Achse im Streckenbetrieb wird der modale Offset der Z-Achse (P160ff) verschoben, was effektiv einer Verschiebung des NC-Koordinatensystems entspricht. Bei Bedarf kann diese Verschiebung mit einer G121 rückgängig gemacht und mit der NC-Istposition (P0ff) verrechnet werden.

Wert	Bedeutung
0	schräggestellte X-Achse ausgeschaltet (default)
15	X-Achse ist um 15° zur senkrechten versetzt
-30	X-Achse ist um -30° zur senkrechten versetzt

Der angegebene Winkel darf zwischen -90° und 90° liegen.

4.15.10 MK_GEWINDE_VMAX

Diese Maschinenkonstante begrenzt die Vorschubgeschwindigkeit [mm/min] der Kopplungsachse bei Gewindeschleifen. Nähere Informationen zur Funktion Gewindeschleifen finden Sie in "der ETC Programmierung NC Betriebssystem" bei der G33.

4.15.11 MK_DW224_255

Diese Maschinenkonstante bietet die Möglichkeit, das Ablaufprogramm der SPS zu beeinflussen. Der Inhalt der 32 Einträge dieser MK wird bei der Übertragung der Maschinenkonstanten in den Datenbereich %MW2.224 bis %MW2.255 der SPS kopiert. Die Bedeutung der Datenworte kann vom Programmierer frei festgelegt werden.

Diese MK kann beispielsweise dazu benutzt werden, unterschiedliche Ausrüstung einer Maschine mit Zusatzaggregaten zu konfigurieren. Um nur eine SPS-Version zu erstellen, kann die Ausrüstung in den MK vermerkt werden und das SPS-Programm entsprechend reagieren.

4.16 Liste der Maschinenkonstanten

Nachfolgend ist die Liste der Maschinenkonstanten abgedruckt, wie sie in die Steuerung geladen wird. Die Liste liegt als ASCII-File vor und kann auf einem PC editiert werden. Der Aufbau der Liste entspricht der vorherigen Beschreibung. Es ist ratsam, alle Maschinenkonstanten mit den notwendigen Werten zu versehen.

Die aufgeführten Schlüsselwerte dürfen nicht verändert werden. Lediglich die zugehörigen Zahlenwerte sind zu verändern oder zu ergänzen.

Die einzelnen Zahlenwerte innerhalb eines Schlüsselwortes werden durch ein ”,” (Komma) abgeschlossen. Der letzte Zahlenwert muss durch ein ”;” (Semikolon) abgeschlossen werden.

Beim Editieren ist darauf zu achten, Kommentarzeichen (”/*”, ”*/”) nicht versehentlich mit zu löschen.

```
/*=============================================================*/
/*         Muster-Maschinenkonstanten V1.74 für ETCxC          */
/*-------------------------------------------------------------*/
/*
** BITTE BEACHTEN
** --------------
**
** Achsreihenfolge:
** Die interne Reihenfolge der Achsen wird über MK_APPLACHSIDX
** festgelegt. Alle achsbezogenen Maschinenkonstanten müssen in
** dieser Reihenfolge konfiguriert werden.
**
** Maschinenkonfiguration:
** Die Maschinenkonstante MK_DW224_255, am Ende dieser Datei,
** enthält wichtige Informationen für die SPS. Näheres steht im
** Kommentar neben der Maschinenkonstante.
**
*/
/*=============================================================*/

/*----------------------------------*/
/*   0. Testeinstellungen           */
/*----------------------------------*/
MK_TEST_OHNEMECHANIK          0;       /* Achsrechner wird bedient, beachtet aber nicht den */
                                       /* Istposzaehler */
MK_SPS_DUMMY                  0;       /* 0 = mit integrierter SPS; 1 = ohne */
```

```
/*-----------------------------------------*/
/*       1. Hardware - Konfiguration       */
/*-----------------------------------------*/
MK_HARDKONF                   -1,      /* Zuordnung Achsrechnerausgaenge */
                              -1,      /* (0-7) zu Achsen; -1 = Achse nicht angeschlossen */
                              -1,
                              -1,
                              -1,
                              -1,
                              -1,
                              -1;

MK_CANDRIVES                   0,      /* Zuordnung Achsen - CAN-Knoten 1-12 */
                               1,
                              -1,
                              -1,
                              -1,
                              -1,
                              -1,
                              -1;

MK_ESABKONF                   -1,      /* Zuordnung der Achsen zu den Achsadressen 1-8 */
                                       /* am ESA-Bus */
                              -1,
                              -1,
                              -1,
                              -1,
                              -1,
                              -1,
                              -1;

MK_APPLACHSIDX                 0,      /* Interne Nummer der X-Achse, */
                               1,      /* Y-Achse */
                               2,      /* Z-Achse */
                               3,      /* C-Achse */
                               4,      /* U-Achse */
                               5,      /* V-Achse */
                               6,      /* W-Achse */
                               7,      /* A-Achse */
                               8,      /* B-Achse */
                               9,      /* u-Achse */
                              10,      /* v-Achse */
                              11,      /* w-Achse */
                              12,      /* x-Achse */
                              13,      /* y-Achse */
                              14,      /* z-Achse */
                              15,      /* a-Achse */
                              16,      /* b-Achse */
                              17;      /* c-Achse */
MK_APPLSPINDELIDX             -1,      /* Zuordn. Spindeln - Achskanälen */
                              -1,      /* -1 = Spindel nicht benutzt */
                              -1;
```

```
MK_ACHSENART                0,      /* Definition der Achse (bit-codiert) */
                            0,      /* Bit 0: 0 Linearachse, 1 Rotationsachse */
                            0,      /* Bit 1: 0 HW-Endschalter beachten, 1 ignorieren */
                            0,      /* Bit 3/2: 00 Normalachse, 01 Spindel */
                                    /* 10 Messachse, 11 Spindel & Messachse */
                            0,      /* Bit 5/4: Einstellung Rotationsachsen (Bit0 = 1) */
                            0;      /* 00 Rotationsachse mit absoluter Positionierung */
                                    /* 01 Modulo 360°-Achse, Vorzeichen gibt Richtung an */
                                    /* 11 Modulo 360°-Achse, kürzester Weg wird gefahren */
                                    /* Bit 7: Handrad */

MK_SPINDELART               1,      /* Zuordnung Spindelhandler-Spindeln */
                            1,      /* 1 = Standard-Spindelhandler */
                            1;


/*-----------------------------------------*/
/*      2.  Software-Konfiguration         */
/*-----------------------------------------*/
MK_KUNDE                    "";     /* aktiviert verschiedene kundenspez. Erweiterungen */

MK_VOREINSTELLUNG           "";     /* für Initialisierung nach Reset oder Programmende */

MK_PASSWORT                 0;      /* Passwort (Nr.) für Maschinenkonstanten-Editor */
                                    /* beim integrierten Bedienteil */

MK_SPRACHE                  0;      /* Sprachauswahl bei integ. Bedienteil; 0=deutsch */

MK_FLOPPYDISK               0;      /* Nr. Schnittstelle an der die Floppy angeschlossen */
                                    /* ist, 0 = keine Floppy */

MK_DRUCKER                  0;      /* Nr. der Schnittstelle an der der Drucker */
                                    /* angeschlossen ist, 0 = kein Drucker */

MK_DRUCKER_V24MODE          9600,   /* Einstellungen für die Druckerschnittstelle */
                            8,      /* Baudrate, Datenbits, Parity, Stopbits */
                            0,
                            1;

MK_CANOPEN_BAUDRATE         0,      /* CAN1-Bitrate für CAN-Open, bzw. 0 für SLIO */
                            1000;   /* CAN2-Bitrate für CAN-Open-Antriebe */

MK_FEHLERRESTART            0;      /* 1 = Wiederanlauf nach Fehler erlaubt */

MK_S0T0_VERSATZ_ERLAUBT     0;      /* 1 = Verschiebung von S0 oder T0 erlaubt */

MK_S_VERSATZSPERRE          0;      /* Nr. des ersten geschützten Werkstückkoordinaten- */
                                    /* Systems oder 0, falls inaktiv */

MK_EDIT_SOFTKEYS            "GXYZMNCFP=IJRKE:P=()ABDHLOQSTUVW{}%";

MK_TEACHSTELLEN             3;      /* Anzahl der Nachkommastellen, die beim */
                                    /* Teachen berücksichtigt werden sollen */

MK_TEACHHEADER              "";     /* Zusatzparameter für geteachte Zeile */
```

```
MK_NCPROG_OHNE_KOMMENTARE     0;       /* 1 = DIN-Programme ohne Komment. in Flash-PROM legen */

MK_NCPROG_NICHT_INS_EEPROM    0;       /* 1 = DIN-Programme nicht ins Flash-PROM ablegen */

MK_NULLPUNKTE_SPEICHERN       0;       /* 1 = Nullpunkte automatisch speichern */
                                       /* (nur wenn CMOS-RAM vorhanden) */

MK_METRISCH                   1;       /* 1 = Weg- und Geschwindigkeitseingaben */
                                       /* im metrischen System [mm] / [mm/min] */
                                       /* 0 = Weg- und Geschwindigkeitseingaben */
                                       /* im inch-System [inch] bzw. [inch/min] */

MK_CONST_REL_MM               1.0;     /* Eingabeaufloesung im metrischen */
                                       /* Masssystem, bezogen auf 1 mm  */

MK_CONST_REL_INCH             25.4;    /* Eingabeaufloesung im inch Masssystem, */
                                       /* bezogen auf 1 mm  */

MK_SW_ENDS_MIT_RAMPE          0;       /* 0 = bei SW-Endschalter ohne Rampe bremsen */
                                       /* 1 = mit Rampe bremsen */

MK_OVERRIDEMAX                1500;    /* max. zulaessiger Override [0.1%] */

MK_EPSILONMM                  0.01;    /* Toleranz bei Positionen (z. B. Kreis) [mm] */

MK_EPSILONGRAD                0.01;    /* Toleranz bei Winkeln [Grad] */

MK_KONTURFEHLER               0.01;    /* maximal zulaessiger Bahnfehler [mm] */

MK_LAH_GRENZWINKEL            90;      /* Übergangswinkel ab dem zwangsweise bis */
                                       /* zum Stillstand gebremst werden soll */

MK_RADIUS_B_BEWERTUNG         1;       /* Beschleunigungsbewertung bei Kreisen oder */
                                       /* Spline-Bögen zur Geschwindigkeitsreduzierung */

MK_HANDRADFILTER              0;       /* Filterzeitkonst. für weiche Handradfunktion in [ms] */

MK_MASCH_POLAR_KART           0;       /* 1 = polares Maschinenkoordinatensystem */

MK_KARTESISCH_ACHSNR          0,       /* Nr. der cos-Achse im kartesischem Koordinatensyst. */
                              1,       /* Nr. der sin-Achse im kartesischen Koordinatensyst. */
                              -1;      /* Nr. der tangentialen Nachführachse oder -1 */
MK_POLAR_ACHSNR               0,       /* Nr. der Radius-Achse im pol. Koordinatens. */
                              1;       /* Nr. der Winkel-Achse im pol. Koordinatens. */

MK_DELTAT                     4;       /* Grobinterpolationstakt in [ms] */

MK_FIT_PRO_GIT                2;       /* Anzahl der Feininterpolationstakte */
                                       /* pro Grobinterpolationstakt */

/* !!! die folgenden 5 MK werden erst nach dem nächsten       !!! */
/* !!! Hochlaufen der Steuerung nach der MK-Übertragung aktiv !!! */
```

```
MK_KANALANZAHL              1;      /* Anzahl der NC-Kanäle */

MK_PFELDGROESSE             2048;   /* Parameterfeldgröße */

MK_LAH_VORLAUFTIEFE         256,    /* Größe Vorlaufpuffer in Sätzen für Kanal 0 */
                            0;      /* und für Kanal 1 */

MK_LAH_RUECKLAUFGRENZE      4,      /* Anz. Sätze, die auf der Kontur zurückgef. */
                            0;      /* werden können für Kanal 0 und Kanal 1 */

MK_SPV_SPEICHERGROESSE      1000;   /* Größe intern. Programmspeichers in kByte */

MK_SPV_SYMBOLANZAHL         0;      /* Max. Anzahl der Symbole für symbolische */
                                    /* Programmnummernverwaltung */
MK_HEADERANZAHL             100;    /* max. Anzahl gleichzeitig zu verwaltender */
                                    /* NC-Programme */
MK_SPS_SPEICHERGROESSE      128;    /* Größe des SPS-Programmspeichers in kByte */

MK_SPS_DATENGROESSE         32;     /* Größe des SPS-Datenspeichers in kByte */
/*------------------------------   -----------*/
/*  3.  Einstellung der Achsen            */
/*------------------------------ ----------*/
/* Aufloesung Weggeber */
/* ---------------------------------------- */
MK_IMPULSE                  65536,  /* Anzahl Impulse pro [MK_WEG] */
                            65536,  /* (nach der Vervierfachung!) */
                            65536,
                            65536,
                            65536,
                            65536;

MK_WEG                      10,     /* Angabe in [mm] oder [Grad]        */
                            10,     /* [WEG] bei dem der Achsrechner (!) */
                            10,     /* Anzahl [MK_IMPULSE] sieht         */
                            10,
                            10,
                            10;
MK_SSIKONF                  0,      /* Konfiguration von SSI-Gebern */
                            0,      /* Bit 0-6: Anz. der Signifikanten Bits (12-32) */
                            0,      /* Bit 7 : 1=Binär-Code, 0=Gray-Code */
                            0,
                            0,
                            0;

MK_ACHSEINGAENGE            1234,   /* Zuordn. der Endschalter zu Achseingängen */
                            1234,   /* 1234   */
                            1234,   /* Eingangsnummer des Reserveeingangs */
                            1234,   /* Eingangsnummer des Referenznockens */
                            1234,   /* Eingangsnummer des neg. Endschalters*/
                            1234;   /* Eingangsnummer des pos. Endschalters*/

MK_US                       0,      /* Überwach. des Weggeberschmutzsignals */
```

```
                        0,      /* 1 = Überwachung aktiv, 0 = inaktiv */
                        0,
                        0,
                        0,
                        0;

MK_UA0                  0,      /* Kabelbruchüberwachung Weggebernullimpuls */
                        0,      /* 1 = Überwachung aktiv, 0 = inaktiv  */
                        0,
                        0,
                        0,
                        0;

MK_UA12                 0,      /* Kabelbruchüberwachung Weggebersignal Ua1/Ua2 */
                        0,      /* 1 = Überwachung aktiv, 0 = inaktiv */
                        0,      /* 2 = einmalige Störung ignorieren */
                        0,
                        0,
                        0;

MK_TEACHACHSE           1,      /* Markierung der Achsen die beim Teachen */
                        1,      /* über die Teach-In-Box übernommen  */
                        1,      /* werden sollen */
                        0,
                        0,
                        0;

MK_MASSSTAB             1,      /* Multiplikationsfaktor für den in einem */
                        1,      /* DIN-Satz programmierten Weg  */
                        1,
                        1,
                        1,
                        1;

MK_HANDRADZUORDNUNG     0,      /* Zuordnung von Achsen zu Handrädern */
                        0,      /* 0 = Handrad 0, 1 = Handrad 1 */
                        0,
                        0,
                        0,
                        0;

MK_HANDRADFAKTOR        1,      /* Individueller Bewertungsfaktor für die */
                        1,      /* Handradfunkt., z.B. zur Sonderbehandlung */
                        1,      /* von Rotationsachsen */
                        1,
                        1,
                        1;

MK_GRUNDOFFSET          0,      /* Abstand des Maschinennullpunkts*/
                        0,      /* vom Referenzpunkt [mm] bzw. [Grad] */
                        0,
                        0,
```

```
                        0,
                        0;

MK_SYNCHRONOFFSET       0,      /* Abstand der Maschinennullpunkte zweier */
                        0,      /* Synchronachsen nach Referenzpunktfahrt */
                        0,
                        0,
                        0,
                        0;

MK_SPINDELUMKEHRSPIEL   0,      /* Spindelumkehrspielkompensation */
                        0,      /* in [mm] bzw. [Grad] */
                        0,
                        0,
                        0,
                        0;

MK_SCHLEPPABSTAND       32000,  /* Maximaler Schleppabstand in */
                        32000,  /* Istwertgeber-Incrementen*/
                        32000,
                        32000,
                        32000,
                        32000;

MK_SCHLEPPZAEHLER       1,      /* Anzahl der Feininterpolationstakte in denen der */
                        1,      /* max. Schleppabstand überschritten werden darf */
                        1,      /* bis einSchleppfehler gemeldet wird */
                        1,
                        1,
                        1;

MK_SCHLEPPGENAUHALT     0.001,  /* Max. Schleppabstand für Genauhalt */
                        0.001,  /* mit Schleppabstandsüberwachung */
                        0.001,  /* in [mm] bzw. [Grad] */
                        0.001,
                        0.001,
                        0.001;

MK_GENAUHALTZEIT        0,      /* Zeit [s] in der der Schleppabstand kleiner */
                        0,      /* als MK_SCHLEPPGENAUHALT sein muss */
                        0,
                        0,
                        0,
                        0;

MK_SYNCHRONABWEICHUNG   10,     /* Max. erlaubte Synchronabweichung von */
                        10,     /* Synchronachsen in [mm] bzw. [Grad] */
                        10,
                        10,
                        10,
                        10;
```

```
MK_SW_ENDS_MINUS            -10000,  /* Software-Endbegrenzung (+)-Richtung */
                            -10000,  /* [mm], bezogen auf Grundoffset  */
                            -10000,
                            -10000,
                            -10000,
                            -10000;

MK_SW_ENDS_PLUS             10000,   /* Software-Endbegrenzung (-)-Richtung */
                            10000,   /* [mm], bezogen auf Grundoffset */
                            10000,
                            10000,
                            10000,
                            10000;

/* ----------------- */
/* Reglereinstellung */
/* ----------------- */

MK_DRIFTABGLEICH            0,       /* Offsetjustierung [mV] für analoge */
                            0,       /* Achsinterfaces */
                            0,
                            0,
                            0,
                            0;

MK_REGLER_MODE              2,       /* 0 = Regelkreis offen */
                            2,       /* 1 = reserviert */
                            2,       /* 2 = P-Regler */
                            2,       /* 3 = P-Regler mit Feed-Forward   */
                            2,       /* 4 = Spindel  */
                            2;       /* 5 = PID-Regler mit Feed-Forward */

MK_KP                       1,       /* KP des P-Reglers */
                            1,       /* (auch für Synchronregelung) */
                            1,
                            1,
                            1,
                            1;

MK_TV                       0,       /* Vorhaltzeit [s] für D-Anteil von */
                            0,       /* PID-Regler (auch für Synchronregelung) */
                            0,
                            0,
                            0,
                            0;

MK_TN                       0,       /* Nachstellzeit [s] für I-Anteil von */
                            0,       /* PID-Regler (auch für Synchronregelung) */
                            0,
                            0,
                            0,
                            0;
```

```
MK_KF                       0,      /* Feed-Forward-Faktor */
                            0,      /* Berechnung: (Umax = Sollwert bei Vmax) */
                            0,      /*                Umax[0.1V] * 327.68                 */
                            0,      /* MK_KF = ------------------------------------------ */
                            0,      /*         Vmax [m/min]   MK_DELTAT     MK_IMPULSE[]   */
                            0;      / *        ------------ * ---------- * -------------   */
                                    /*            60[]      MK_FIT_PRO_GIT    MK_WEG[mm]   */

MK_KB                       0,      /* Faktor für Beschleunigungsvorsteuerung */
                            0,
                            0,
                            0,
                            0,
                            0;

MK_T2                       0,      /* Filterzeitkonstante für Feininterpolation [s] */
                            0,
                            0,
                            0,
                            0,
                            0;



/* ------------------------------------ */
/* Einstellungen fuer Referenzpunktfahrt */
/* ------------------------------------ */
MK_REF_TYP                  0,      /* Art der Referenzpunktfahrt */
                            0,      /* 0: mit Nocken und Nullimpuls, */
                            0,      /* 1: mit Nocken aber ohne Nullimpuls, */
                            0,      /* 2: nur Zaehler nullen mit G74, */
                            0,      /* 3: nur Nullimpuls, Nocken gibt Richtung an */
                            0;      /* 4: 2 Nocken, Reserve-Eingang am ARI ist zusätzlicher*/
                                    /* Referenznocken und wird zuerst angefahren */

MK_REF_RICHTUNG_UND_FOLGE   1,      /* Richtung Achse bei Start Ref.-Punktfahrt */
                            1,      /* und Reihenfolge bei auto. Ref.-Punktfahrt */
                            1,      /*  0 => Achse ist nicht beteiligt */
                            1,      /*  X => pos. Richtung */
                            1,      /* -X => neg. Richtung */
                            1;

MK_REF_VMAX1                5,      /* Geschwindigkeit zum Nocken [m/min] bzw. [U/min] */
                            5,
                            5,
                            5,
                            5,
                            5;

MK_REF_BMAX1                1,      /* Rampe für Fahrt zum Nocken [m/sec2] bzw. [U/sec2] */
                            1,
                            1,
```

```
                        1,
                        1,
                        1;

MK_REF_VMAX2            1,      /* Geschwindigkeit vom Nullimpuls [m/min] bzw. [U/min] */
                        1,
                        1,
                        1,
                        1,
                        1;

MK_REF_BMAX2            1,      /* Rampe bei Fahrt vom Nocken [m/sec2] bzw. [U/sec2] */
                        1,
                        1,
                        1,
                        1,
                        1;



/* ----------------------- */
/* Achsbezogene Grenzwerte */
/* ----------------------- */

MK_MODVMAX              10,     /* modale Achsgeschwindigkeit */
                        10,     /* (Handverfahren) [m/min] bzw. [U/min] */
                        10,
                        10,
                        10,
                        10;

MK_VMAX                 20,     /* max. Achsgeschw. [m/min] / [U/min] */
                        20,
                        20,
                        20,
                        20,
                        20;

MK_BESCHL               2,      /* Beschleunigungsrampe [m/sec2] / [U/sec2] */
                        2,
                        2,
                        2,
                        2,
                        2;

MK_BREMS                2,      /* Bremsrampe [m/sec^2] bzw. [U/sec^2] */
                        2,
                        2,
                        2,
                        2,
                        2;
```

```
MK_T_BESCHL                 0,      /* Dämpfungszeitkonstante für Beschleunigungs- */
                            0,      /* und Bremsrampen [ms] */
                            0,
                            0,
                            0,
                            0;

MK_SPINDELMAX               6000,   /* Spindeldrehzahl in [U/min] bei 10V */
                            6000,
                            6000;
```

```
/* ----------------------- */
/* Bahnbezogene Grenzwerte */
/* ----------------------- */
MK_VBAHNMAX                 20;     /* 20 max. Bahngeschwindigkeit [m/min] */

MK_BAHNBESCHL               2;      /* Beschleunigungsrampe [m/sec2] */

MK_BAHNBREMS                2;      /* Bremsrampe [m/sec2] */

MK_T_BAHNBESCHL             0;      /*Dämpfungszeitkonst. Brems-/Beschleunigungsrampen [ms]*/

/*-----------------------------------------*/
/*  4.  Einstellung Zusatzkarten           */
/*-----------------------------------------*/
MK_MESS_AUFLOESUNG          1.0,    /* Auflösung der Analog-Messwerte [mm/V] */
                            1.0,
                            1.0,
                            1.0,
                            1.0,
                            1.0,
                            1.0,
                            1.0,
                            1.0,
                            1.0,
                            1.0,
                            1.0,
                            1.0,
                            1.0,
                            1.0,
                            1.0,
                            1.0;
/*-----------------------------------------------*/
/*  5.  Technologie - spezifische Einstellungen  */
/*-----------------------------------------------*/
MK_MFKT_UPR_TABELLE         0,      /* Tabelle der M-Funktionen nach denen eine */
```

```
                        0,      /* G22 L9000+Mfktnr eingefügt werden soll.  */
                        0,      /* Die Tabelle darf max. 16 Einträge haben  */
                        0,
                        0,
                        0,
                        0,
                        0,
                        0,
                        0,
                        0,
                        0,
                        0,
                        0,
                        0,
                        0;
MK_WLK_C_GRENZWINKEL    45;     /* Schneiden/Nähen: Winkel bis zu dem die */
                                /* C-Achse schlagartig zugestellt werden soll */

MK_X_WINKEL             0;      /* Winkel der schräggestellten X-Achse in */
                                /* Grad (X-Z Ebene) */

MK_GEWINDE_VMAX         0;      /* Gewindeschleifen */

MK_SPINDELMAX_GS1       6000,   /* Spindeldrehzahl in [U/min] bei 10 V */
                        6000,   /* für Getriebestufe 1 für Spindeltyp 9 */
                        6000;

MK_SPINDELMAX_GS2       6000,   /* Spindeldrehzahl in [U/min] bei 10 V */
                        6000,   /* für Getriebestufe 2 für Spindeltyp 9 */
                        6000;

MK_SPINDELMAX_GS3       6000,   /* Spindeldrehzahl in [U/min] bei 10 V */
                        6000,   /* für Getriebestufe 3 für Spindeltyp 9 */
                        6000;

MK_SPINDELDREHZAHLMAX   6000,   /* Max. zulässige Spindeldrehzahl in [U/min] */
                        6000,   /* (nicht bei allen Spindelhandlern */
                        6000;   /* berücksichtigt.) */

MK_SPINDELDREHZAHLMIN   0,      /* Min. zulässige Spindeldrehzahl in [U/min] */
                        0,      /* (nicht bei allen Spindelhandlern berücksichtigt) */
                        0;

MK_TECHNOLOGIEDATEN1    0,      /* Anwendungsspez. Technologieparam., die */
                        0,      /* ab P760 im Parameterfeld liegen */
                        0,
                        0,
                        0,
                        0,
                        0,
                        0,
                        0,
                        0;
```

```
MK_TECHNOLOGIEDATEN2         0,       /* wie TECHNOLOGIEDATEN1 */
                             0,
                             0,
                             0,
                             0,
                             0,
                             0,
                             0,
                             0,
                             0;
MK_TECHNOLOGIEDATEN3         0,       /* wie TECHNOLOGIEDATEN1 */
                             0,
                             0,
                             0,
                             0,
                             0,
                             0,
                             0,
                             0,
                             0;

MK_TECHNOLOGIEDATEN4         0,       /* wie TECHNOLOGIEDATEN1 */
                             0,
                             0,
                             0,
                             0,
                             0,
                             0,
                             0,
                             0,
                             0;

/*-------------------------------------------*/
/*  6.  SPS - spezifische Einstellungen      */
/*-------------------------------------------*/

/* Die folgenden Werte werden in den Datenbaustein 2 der SPS geschrieben */
MK_DW224_255                 0,       /* DW 224 */
                             0,       /* DW 225 */
                             0,       /* DW 226 */
                             0,       /* DW 227 */
                             0,       /* DW 228 */
                             0,       /* DW 229 */
                             0,       /* DW 230 */
                             0,       /* DW 231 */
                             0,       /* DW 232 */
                             0,       /* DW 233 */
                             0,       /* DW 234 */
                             0,       /* DW 235 */
                             0,       /* DW 236 */
                             0,       /* DW 237 */
```

```
0,      /* DW 238 */
0,      /* DW 239 */
0,      /* DW 240 */
0,      /* DW 241 */
0,      /* DW 242 */
0,      /* DW 243 */
0,      /* DW 244 */
0,      /* DW 245 */
0,      /* DW 246 */
0,      /* DW 247 */
0,      /* DW 248 */
0,      /* DW 249 */
0,      /* DW 250 */
0,      /* DW 251 */
0,      /* DW 252 */
0,      /* DW 253 */
0,      /* DW 254 */
0;      /* DW 255 */
```

5 Schnittstelle SPS <-> NC-Betriebssystem

Die integrierte SPS ist Bestandteil der Hard- und Software der Steuerung ETC. Sie wird wie eine herkömmliche SPS programmiert und hat im Wesentlichen die gleichen Eigenschaften. Zur Kommunikation mit der eigentlichen CNC-Steuerung (NC) existiert innerhalb der CNC eine formale Schnittstelle, die in ihrer Funktion an den Vorschriften nach IEC 550, ISO 4336 und VDI 3422 orientiert ist.

In diesem Kapitel wird die Funktion der Schnittstelle beschrieben. Die Einzelsignale werden definiert und ihre Wirkungsweise erläutert.

Eine Beschreibung der Signalformen sowie des Timings der Signale ist nicht erforderlich, da die Schnittstelle lediglich durch interne Speicher gebildet wird und somit den gleichen Regeln wie der Datenaustausch innerhalb der CNC unterliegt. Signallaufzeiten müssen nicht berücksichtigt werden.

5.1 Definitionen

Der Datenaustausch zwischen NC-Rechner und integrierter SPS erfolgt über einen internen RAM-Bereich, der als ”Datenworte” für die SPS festgelegt ist. Die Übergabe erfolgt in den Datenbausteinen ”DB0”, ”DB1”, ”DB2” und ”DB8 ... DB15”.

DB0: Enthält die Schnittstelle zum ETC_MMI.

DB1: Enthält alle Standardsignale (NC <-> SPS) sowie einen Bereich aus der ”virtuellen Tastatur”.

DB2: Schnittstelle zwischen SPS und MMI.

DB8 ... DB15: Hier werden die Schnittstellendaten zwischen NC-Rechner und MMI abgelegt. Diese Schnittstelle entspricht der Dual-Port RAM Schnittstelle, die der Anbindung an das MMI dient.

Jeder Datenbaustein umfasst 256 Datenworte (DW) zu 16 Bit. In einem Baustein sind sowohl statische Signale als auch Nachrichten (Daten), die mit Quittungsmerker (”handshake”) ausgetauscht werden, enthalten. Die Datenbereiche sind bidirektional, d. h. sowohl die Daten von der SPS an den NC, als auch diejenigen vom NC an die SPS, sind zusammengefasst.

Hinweis!

Im CoDeSys stehen die Signale/Daten der Datenbausteine als sogenannte Systemvariablen zur Verfügung.

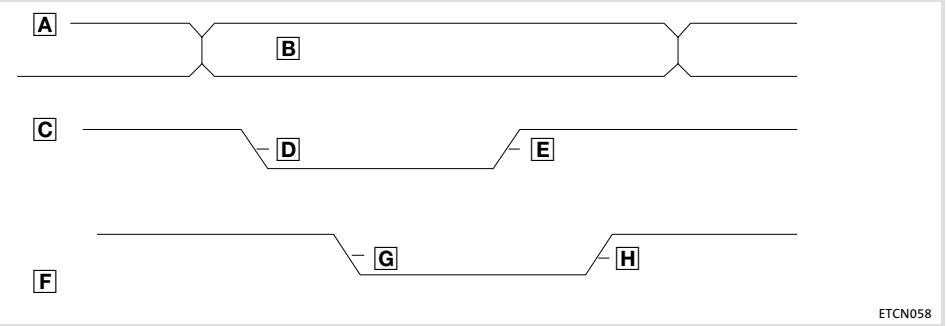
Der Datenaustausch zwischen NC und SPS erfolgt zyklisch bei jedem Durchlauf des SPS-Programms. Die Daten, die vom NC an die SPS übertragen werden, werden *vor* Beginn eines SPS-Durchlaufs kopiert, die Daten von der SPS an den NC, *nach* einem Durchlauf.

Statische Signale werden gesetzt, ohne zuvor deren Auswertung nachzufragen. Daten, deren jeweilige Auswertung sichergestellt werden muss, wer-

den als Nachricht übertragen und haben zusätzlich zwei Signale für einen handshake: ”Strobe” und ”Quittung”.

Mit ”Strobe” setzt der Sender die zuvor geschriebenen Daten gültig, mit ”Quittung” signalisiert der Empfänger die Auswertung der Daten. Erkennt der Sender die Quittierung des Empfängers, nimmt er seinerseits das ”Strobe”-Signal weg, woraufhin der Empfänger sein ”Quittung”-Signal löscht. Erst jetzt ist eine Übergabe neuer Daten erlaubt. Bei der Übergabe der Daten sind keine besonderen zeitlichen Bedingungen zu beachten, lediglich die Reihenfolge der Signale muss in der beschriebenen Weise erfolgen. Die Signale ”Strobe” und ”Quittung” werden zeitlich überwacht. Sie dürfen maximal für einen Zeitraum von einer Sekunde stehen bleiben. Wird der Zeitraum überschritten, erfolgt die Ausgabe einer Fehlermeldung, die dazu führt, dass ein laufendes DIN-Programm abgebrochen wird.

Prinzipiell erfolgt der Datenaustausch bei Nachrichten immer nach folgendem Signalschema:



A **Daten**
B Neuer Wert
C **Strobe**
D Gültig
E Übernahme erkannt
F **Quittung**
G Neuen Wert übernommen
H Nächster Wert erkannt

In der nachfolgenden Beschreibung der Signale und Daten ist die jeweilige Signalart angegeben als

► statisch

► Nachricht

► Strobe

► Quittung

Die Bezeichnung der Datenworte erfolgt im Weiteren nach folgendem Schema:

Dxxx.yy

xxx = Nummer des Datenworts in einem Datenbaustein (0 ... 255)

yy = Bit - Nummer in einem Datenwort (0 ... 15)

5.1.1 Datenbaustein 0

Bei der Verwendung des DB0 ist zu beachten, dass die Daten nur in der 2. SPS-Task (PLC_PRG2/ OB20) ausgewertet werden dürfen. Diese Notwendigkeit ergibt sich aus der Tatsache, dass die Daten im DB0 vor dem Aufruf der 2. Task aktualisiert werden und der NC nach der Ausführung die Daten auswertet. Eine Ausnahme ist die Fehlerschnittstelle, da sie durch ein Strobe- (Fehlerzähler) und Quittungssignal (Fehlerquittung) synchronisiert ist.

Belegung

Datenwort	CoDeSys Systemvariable	Richtung
001.00 – 001.15	db0_sps2nc_nc_fehlerquittung_w	SPS → NC
016.00 – 095.15	db0_nc2sps_fehlertext_s	SPS ← ß C
096.00 – 127.15	db0_nc2sps_fehlerinfo_s	SPS ← NC
128.00 – 128.15	db0_nc2sps_aktuelles_menu_w	SPS ← NC
129.00 – 129.15	db0_nc2sps_ob20_ereignis_w	SPS ← NC
132.00 – 132.15	db0_nc2sps_funktionsstatus_w	SPS ← NC
133.00 – 133.15	db0_nc2sps_nc_fehlerzaehler_w	SPS ← NC
134.00 – 134.15	db0_nc2sps_nc_fehlerklasse_w	SPS ← NC
135.00 – 135.15	db0_nc2sps_nc_fehlermodul_w	SPS ← NC
136.00 – 136.15	db0_nc2sps_nc_fehlernummer_w	SPS ← NC
137.00 – 137.15	db0_nc2sps_aktuelles_untermenu_w	SPS ← NC
148.00 – 255.15	db0_nc2sps_g253_textausgabe_s	SPS ← NC

Alle nicht explizit aufgeführten Bereiche sind reserviert.

Beschreibung der Signale

Datenwort	Name	Richtung	Signalart
001	Fehlerquittung	SPS → NC	statisch

Dieses Datenwort kann von der SPS zur Erkennung von Fehlermeldungen und zum Quittieren von Fehlermeldungen benutzt werden, falls es notwendig ist, dass die SPS eine eventuell ausgegebene Fehlermeldung auswerten muss. Ist DW001 ungleich DW133 (Fehlerzähler), dann liegt eine neue Fehlermeldung vor. Hat die SPS die Fehlermeldung ausgewertet, dann muss DW001 auf den Wert von DW133 gesetzt werden. Damit wird dem NC-Rechner der Fehler quittiert.

Datenwort	Name	Richtung	Signalart
016.00 – 095.15	Fehlertext	NC → SPS	statisch

Ab DW016 wird der Fehlertext des aktuell anstehenden Fehlers als nullterminierter String abgelegt.

Datenwort	Name	Richtung	Signalart
096.00 – 127.15	Fehlerzusatzinformation	NC → SPS	statisch

Ab DW096 wird eine evt vorhandene Zusatzinformation (z. B. Achsbuchstabe) zum aktuell anstehenden Fehler als nullterminierter String abgelegt.

Datenwort	Name	Richtung	Signalart
128	Aktuelles Menu	NC → SPS	statisch

Hier steht eine Kennung des aktuellen Menüs:

0 = ungültig
1 = Hauptmenü
2 = Einrichten
3 = Automatik
4 = Programmieren
5 = Diagnose
6 = Benutzer

Soll vom SPS-Programm ein Menüwechsel zu einem Standardmenü vorgenommen werden, so muss vor Ausführung der Funktion "Neuer_Zustand()" die Kennung des neuen Menüs in DW128 abgelegt werden.

Datenwort	Name	Richtung	Signalart
129	Ereignis	NC → SPS	statisch

Hier legt der NC vor dem Aufruf der 2. SPS-Task (PLC_PRG2/ OB20) eine Kennung ab, aus welchem Grund die 2. SPS-Task ausgeführt werden soll, so dass dort entsprechend reagiert werden kann. Der Inhalt wird in jedem SPS-Zyklus aktualisiert und muss nicht zurückgesetzt werden.

0 = zyklischer Aufruf
1 = Tastenmeldung => DW130 auswerten
2 = Eingabeabschluss => DW131 auswerten
3 = neue Maschinenkonstanten wurden geladen
4 = Tastenmeldung bei aktiver Eingabe => DW130 auswerten
5 = Satzvorlaufgrenze erreicht
6 = Satzvorlaufzielposition erreicht

Datenwort	Name	Richtung	Signalart
132	Status nach Funktionsende	NC → SPS	statisch

Jede aufgerufene Funktion (Ba_Einrichten(), Ba_Automatik(), usw.) gibt hier einen Status zurück:

0 = Betriebsart noch aktiv.
1 = Betriebsart nicht mehr aktiv. System muss in den Grundzustand (Hauptmenü) gebracht werden.

Dieser Status muss nach Ablauf der Funktion vom SPS-Programm überprüft werden. Hat DW132 den Wert 1, bedeutet das, dass in der zuvor abgelaufenen Funktion in den Grundzustand geschaltet wurde.

Datenwort	Name	Richtung	Signalart
133	Fehlerzähler	NC → SPS	statisch

DW133 wird vom NC-Rechner benutzt, um Fehler an die SPS zu melden. Dazu wird DW133 inkrementiert, und ist somit ungleich DW001 (Fehlerquittung). Nach Auswertung der Fehlermeldung muss DW001 von der SPS ebenfalls inkrementiert werden. Daran erkennt der NC-Rechner die Fehlerquittierung und kann gegebenenfalls eine weitere Fehlermeldung absetzen.

Datenwort	Name	Richtung	Signalart
134	Fehlerklasse	NC → SPS	statisch

Hier wird für jeden gemeldeten Fehler die Fehlerklasse mitgeteilt. Es gibt folgende Fehlerklassen:

1 = leichter lokaler Fehler
2 = schwerer lokaler Fehler
3 = leichter globaler Fehler
4 = schwerer globaler Fehler

Lokale Fehler betreffen nur das Sub-System, das den Fehler gemeldet hat. Globale Fehler haben immer ein Anhalten bzw. Unterbrechen des gerade ablaufenden DIN-Programmes zur Folge.

Datenwort	Name	Richtung	Signalart
135	Fehlermodul	NC → SPS	statisch

Hier wird für jeden gemeldeten Fehler das Modul mitgeteilt, das den Fehler gemeldet hat. Folgende Module können Fehler melden:

1 = Achsrechner
2 = Initialisierung
3 = Interpreter
4 = Kopplung
5 = PLC-Laufzeitsystem
6 = Grobinterpolator
7 = Speicherverwaltung
9 = Zentralsteuerung
10 = SPS
11 = Betriebssystem
12 = CAN
13 = Hilfsroutinen
14 = Kommunikation NC -> MMI
15 = Kommunikation MMI -> NC
18 = DS402 konformer Antriebsverstärker
19 = Lenze Antriebsverstärker

Datenwort	Name	Richtung	Signalart
136	Fehlernummer	NC → SPS	statisch

Hier wird die eigentliche Fehlernummer gemeldet. Zur Fehlernummer gehört immer das Fehlermodul. Nur über Fehlernummer **und** Fehlermodul kann der Fehler eindeutig bestimmt werden.

Datenwort	Name	Richtung	Signalart
137	Aktuelles Untermenue	NC → SPS	statisch

reserviert

Datenwort	Name	Richtung	Signalart
148.00-255.15	Textausgabe G253	NC → SPS	statisch

Die Texte, die mit der G-Funktion G253 (ohne Parameter E) ausgegeben werden, werden ab DW148 für die SPS eingeblendet.

5.1.2 Datenbaustein 1

Im Datenbaustein 1 werden die Standardsignale ausgetauscht. Unter Standardsignalen versteht man alle in den jeweiligen Vorschriften festgehaltenen Signale und Nachrichten, die den Zustand der SPS und des NC wiedergeben sowie einen Einfluss auf die Arbeit des NC bzw. der SPS haben. Dies sind auch Signale/Nachrichten, die nicht in allgemeinen Vorschriften existieren, jedoch zum Standardumfang der ETC gehören.

Nachfolgend wird eine Übersicht über die Belegung des Datenbausteins gegeben. Daran anschließend werden die Signale und Nachrichten im Einzelnen beschrieben.

Belegung

Datenwort	Name	Richtung
000.00	db1_sps2nc_notaus_bit	SPS → NC
000.01	db1_sps2nc_vorschubfreigabe_bit	SPS → NC
000.02	db1_sps2nc_soforthalt_bit	SPS → NC
000.03	db1_sps2nc_einzelfunktionsperre_bit	SPS → NC
001.00 – 001.15	db1_sps2nc_vorschubfreigabe_w	SPS → NC
002.00 – 002.15	db1_sps2nc_position_halt_w	SPS → NC
003.00 – 003.15	db1_sps2nc_verfahrtastenfreigabe_plus_w	SP → NC
004.00 – 004.15	db1_sps2nc_verfahrtastenfreigabe_minus_w	SPS → NC
005.00 – 005.15	db1_sps2nc_reglerfreigabe_w	SPS → NC
007.00	db1_sps2nc_einlesefreigabe_bit	SPS → NC
009.00 – 009.07	db1_sps2nc_programmstart_b	SPS → NC
009.08 – 009.11	db1_sps2nc_use_spsoverride_b	SPS → NC
012.00	db1_sps2nc_programmstop_b	SPS → NC
012.08	db1_sps2nc_unterbrechen_bit	SPS → NC
013.00 – 013.07	db1_sps2nc_einzel_folgesatz_bit	SPS → NC
013.08 – 013.15	db1_sps2nc_satzausblenden_bit	SPS → NC
014.00	db1_sps2nc_rueckzug_bit	SPS → NC
016.00 – 017.15	db1_sps2nc_tastensignale_aw	SPS → NC
018.00 – 021.15	db1_sps2nc_qbit_signale_aw	SPS → NC
031.00 – 031.15	DB1_sps2nc_freigaben_mmi_w	SPS → NC
032.00	db1_sps2nc_mfkt_quitt_bit	SPS → NC
033.00	db1_sps2nc_hfkt_quitt_bit	SPS → NC
035.00	db1_sps2nc_extsync_enable_bit	SPS → NC
035.08	db1_sps2nc_programmhalt_aktiv_bit	SPS → NC
036.00 – 037.15	db1_sps2nc_qin_mask_aw	SPS → NC
038.00 – 039.15	db1_sps2nc_qout_mask_aw	SPS → NC
042.00 – 043.15	db1_sps2nc_qin_offset_ab	SPS → NC
044.00 – 045.15	db1_sps2nc_qout_offset_ab	SPS → NC
080.00 – 080.15	db1_sps2nc_spsoverride_allg_w	SPS → NC
081.00 – 081.15	db1_sps2nc_spsoverride_spindel_w	SPS → NC
082.00 – 082.15	db1_sps2nc_spsoverride_ozillation_w	SPS → NC
083.00 – 083.15	db1_sps2nc_spsoverride_sps_w	SPS → NC
084.00 – 091.15	db1_sps2nc_verfahr_ab	SPS → NC
128.00	db1_nc2sps_betriebsbereit_1_bit	SPS ← NC
128.01	db1_nc2sps_betriebsbereit_2_bit	SPS ← NC
129.00 – 129.15	db1_nc2sps_referenzpunkt_w	SPS ← NC
130.00 – 131.15	db1_nc2sps_referenzfahrt_laeuft_w	SPS ← NC
131.00 – 131.15	db1_nc2sps_betriebsart_w	SPS ← NC

Datenwort	Name	Richtung
132.00	db1_nc2sps_nc_programm_laeuft_bit	SPS ← NC
132.02	db1_nc2sps_satzvorlauf_aktiv_bit	SPS ← NC
132.03	db1_nc2sps_einzelsatz_laeuft_bit	SPS ← NC
132.04	db1_nc2sps_unterbrechen_aktiv_bit	SPS ← NC
132.08	db1_nc2sps_alle_achsen_stehen_bit	SPS ← NC
133.00 – 133.07	db1_nc2sps_wahlweise_halt_b	SPS ← NC
133.08 – 133.15	db1_nc2sps_programmstartzaehler_b	SPS ← NC
134.00 – 134.15	db1_nc2sps_achse_faehrt_w	SPS ← NC
135.00 – 135.15	db1_nc2sps_modal_am_ziel_w	SPS ← NC
137.00 – 137.07	db1_nc2sps_programmstart_b	SPS ← NC
137.08 – 137.15	db1_nc2sps_einzel_folgesatz_bit	SPS ← NC
138.00 – 138.15	db1_nc2sps_canmodul_vorhanden_w	SPS ← NC
139.00 – 139.15	db1_nc2sps_can_status_w	SPS ← NC
142.00	db1_nc2sps_programmstop_b	SPS ← NC
144.00	db1_nc2sps_vorschub_halt_b	SPS ← NC
150.00	db1_nc2sps_mfkt_strobe_bit	SPS ← NC
151.00 – 151.15	db1_nc2sps_mfkt_w	SPS ← NC
159.00	db1_nc2sps_hfkt_strobe_bit	SPS ← NC
160.00 – 160.15	db1_nc2sps_hfkt_w	SPS ← NC
177.00 – 177.15	db1_nc2sps_refpunkt_angefahren_w	SPS ← NC
178.00 – 178.15	db1_nc2sps_endschalter_plus_w	SPS ← NC
179.00 – 179.15	db1_nc2sps_endschalter_minus_w	SPS ← NC
180.00 – 180.15	db1_nc2sps_referenznocken_w	SPS ← NC
181.00 – 181.15	db1_nc2sps_reserveeingang_w	SPS ← NC
182.00 – 182.15	db1_nc2sps_slave_endschalter_plus_w	SPS ← NC
183.00 – 183.15	db1_nc2sps_slave_endschalter_minus_w	SPS ← NC
184.00 – 184.15	db1_nc2sps_slave_referenznocken_w	SPS ← NC
185.00 – 185.15	db1_nc2sps_slave_reserveeingang_w	SPS ← NC
192.00 – 199.15	db1_nc2sps_copmodul_vorhanden_aw	SPS ← NC
200.00 – 200.15	db1_nc2sps_spsoverride_allg_w	SPS ← NC
201.00 – 201.15	db1_nc2sps_spsoverride_spindel_w	SPS ← NC
202.00 – 202.15	db1_nc2sps_spsoverride_ozillation_w	SPS ← NC
203.00 – 203.15	db1_nc2sps_spsoverride_sps_w	SPS ← NC
204.00 – 211.15	db1_nc2sps_verfahr_ab	SPS ← HMI
212.00 – 219.15	db1_nc2sps_mmi_tasten_aw	SPS ← HMI

Alle nicht explizit aufgeführten Bereiche sind reserviert.

Beschreibung der Signale

Datenwort	Name	Richtung	Signalart
000.00	NOT AUS	SPS → NC	statisch

0 = NOT-AUS Zustand
1 = normaler Betriebszustand

Wirkung im NC: Das 0-Signal unterbricht alle Bewegungen, die Abarbeitung eines Programms wird abgebrochen.

Datenwort	Name	Richtung	Signalart
000.01	Vorschubfreigabe (gesamt)	SPS → NC	statisch
001.00-15	Vorschubfreigabe Achse 0 ... 15	SPS → NC	statisch

Für jede Achse wird ein Signal ”Vorschubfreigabe” ausgegeben. Außerdem existiert ein gemeinsames Freigabe-Signal. Die resultierende Vorschubfreigabe wird als logische UND-Verknüpfung mit den achsabhängigen Vorschubfreigaben gebildet.

Signalzustand ”1” erteilt die Vorschubfreigabe.

Wirkung im NC: Das ”1”-Signal erlaubt eine Bewegung der jeweiligen Achse, wenn nicht das Signal ”Vorschub Halt” gesetzt ist.

Ein ”0”-Signal bewirkt den Stillstand der jeweiligen Achse bzw. aller Achsen. In Bewegung befindliche Achsen werden mit der eingestellten Bremsrampe angehalten.

Bei Interpolation werden alle beteiligten Achsen am Anfahren gehindert bzw. gebremst, wenn die Vorschubfreigabe für eine der beteiligten Achsen verweigert wird.

Im Handbetrieb werden die Verfahraufträge abgebrochen, d. h. auch nach Wiederkehr der Vorschubfreigabe wird ohne neuen Auftrag nicht gefahren.

Im Automatikbetrieb werden die Bewegungen nur unterbrochen, der Verfahrauftrag wird fortgesetzt bei Wiederkehr der Vorschubfreigabe.

Datenwort	Name	Richtung	Signalart
000.02	Sofort-Stop	SPS → NC	statisch

Ein für alle Achsen gemeinsames Signal:

0 = Das Fahren von Achsen wird freigegeben.
1 = Fahrende Achsen werden *ohne* Bremsrampe angehalten (”Sofort-Stop”).

Wirkung im NC: Das ”1”-Signal verursacht, dass die fahrenden Achsen ohne Bremsrampe angehalten werden. Die Bewegungen werden nur unterbrochen, der Verfahrauftrag wird fortgesetzt, wenn das Signal wieder auf den Wert 0 gesetzt wird.

Anwendung: Reaktion auf ”Not Aus” in Anwendungen, bei denen nach einem Not Aus das Programm fortgesetzt werden soll. In diesem Fall wird von der SPS nur die Vorschubfreigabe weggenommen. Sollen die Achsen trotzdem schlagartig stehen bleiben, kann vorher das Signal ”Sofort-Stop” gesetzt werden.

Datenwort	Name	Richtung	Signalart
000.03	MMI Einzelfunktionssperre	SPS → NC	statisch

Mit dem Signal lassen sich alle Einzelfunktionen (z. B. G-Funktionen, M-Funktionen, ...) vom HMI sperren.

Wirkung im NC: Das ”1”-Signal bewirkt, dass kein Einzelsatz vom MMI durch den NC ausgeführt wird.

Anwendung: Das Signal könnte verwendet werden, um Verfahrfunktionen, die über einen Einzelsatz am HMI ausgelöst werden, zu sperren, solange noch keine Referenzpunkte der Achsen bekannt sind.

Datenwort	Name	Richtung	Signalart
002.00-15	Position Halt Achse 0 – 15	SPS → NC	statisch

Für jede Achse kann ein Signal ausgegeben werden, um einen Stillstand der jeweiligen Achse zu bewirken. Bei Signalzustand ”1” hält die Achse in ihrer Position, bei Signalzustand ”0” ist die Positionierung erlaubt.

Das Signal ist standardmäßig auf ”0” gesetzt.

Wirkung im NC: Signalzustand ”1” bewirkt, dass die jeweilige Achse, unabhängig von der Vorschubfreigabe, nicht positioniert wird. Steuerungsintern wird die Achse so behandelt, als ob sie fährt. Lediglich die tatsächliche Positionierung wird unterdrückt. Die Istwerte werden intern erzeugt.

Bei Signalzustand ”0” wird die Achse ohne Einschränkung, abhängig von den übrigen Bedingungen, bewegt. Das Signal ist achsspezifisch, die anderen Achsen sind nicht betroffen. Das gilt auch im Bahnbetrieb.

Das Signal sollte nur bei stehender Achse seinen Zustand wechseln. Bei sich bewegenden Achsen wird die Bewegung sofort ohne Rampe angehalten.

Anwendung: Testbetrieb, bei dem sich bestimmte Achsen nicht bewegen dürfen.

Datenwort	Name	Richtung	Signalart
003	Freigabe Verfahrtasten (+) Achse 0 - 15	SPS → NC	statisch
004	Freigabe Verfahrtasten (-) Achse 0 - 15	SPS → NC	statisch

Es wird für jede Achse ein Freigabe-Signal für die Handverfahrtasten ausgegeben. Signalzustand ”1” erteilt die Freigabe. Das Signal wird nur im ”Handbetrieb” oder ”Unterbrechen” gegeben, wenn alle Bedingungen für ein manuelles Verfahren der Achsen gegeben sind. Das Signal ist unabhängig von Vorschub- oder Reglerfreigabe.

Wirkung in NC: Der Auftrag der Handverfahrtaste (vgl. Beschreibung DB15) wird nur ausgeführt, wenn das Freigabesignal der SPS gegeben ist. Eine positive Flanke des Signals während der Ausführung eines Befehls hat keine Auswirkung. Eine negative Flanke bricht die Ausführung ab.

Anwendung: Das Signal ist von Bedeutung, wenn die Verfahrtasten nicht über die SPS realisiert sind sondern über die ”virtuelle Tastatur” (Bedienrechner).

Mit den Freigaben wird die Wirksamkeit der Verfahrtasten gesteuert. Beim Wert 0 werden die aus Datenbaustein 1 und Datenwort 84 ... 91 Verfahrtasten von der SPS berücksichtigt. Beim Wert 1 werden die über die ”virtuelle Tastatur” realisierten Verfahrtasten (MMI) berücksichtigt.

Datenwort	Name	Richtung	Signalart
005.00-15	Reglerfreigabe Achse 0 - 15	SPS → NC	statisch

Es wird für jede Achse ein Signal ”Reglerfreigabe” ausgegeben. Signalzustand ”1” erteilt die Reglerfreigabe. Die Reglerfreigabe wird vor der Erteilung der Vorschubfreigabe ausgegeben.

Wirkung im NC: Im ”0”-Zustand wird der Lageregelkreis der betreffenden Achse geöffnet. Laufende Bewegungen und die Bearbeitung des Programms wird abgebrochen, die Achsen werden sofort angehalten (ohne Rampe).

Im ”1”-Zustand sind die Regelkreise geschlossen. Beim Schließen der Regelkreise werden die aktuellen Ist-Positionen der Achsen in den NC-Rechner übernommen.

Datenwort	Name	Richtung	Signalart
007.00	Einlesefreigabe	SPS → NC	statisch

Das ”1”-Signal erlaubt die Ausführung des nächsten Satzes.

Wirkung im NC: Wenn das Signal den Zustand ”1” hat, ist die Ausführung des nächsten NC-Satzes erlaubt.

Zustand ”0” verbietet die Bearbeitung des nächsten Verfahrsatzes in der Steuerung (andere Sätze, wie Arithmetiksätze werden bearbeitet!) sowie die Ausgabe von synchronisierten M-Funktionen an die SPS.

Der Wechsel in den Zustand ”0” hat nicht den Abbruch des laufenden Satzes zur Folge.

Im Handbetrieb wird die Bearbeitung eines neuen Datensatzes sowie die Übergabe von M-Funktionen verhindert.

Datenwort	Name	Richtung	Signalart
009.00-07	Programmstart	SPS → NC	statisch

Das Startsignal ist eine Byte-Information, die einen Wert zwischen 0 und 255 annehmen kann.

Die Programmstartanforderung wird ausgegeben, wenn das Startsignal des HMI’s an die SPS (DW137) einen Wert ungleich ”0” aufweist oder ein entsprechender Eingang der SPS gesetzt ist (”Betätigen der Starttaste”) und die notwendigen Startvoraussetzungen gegeben sind.

Der Wert wird auf ”0” gesetzt bei Erkennen des Signals ”NC-Programm läuft”.

Die Information gliedert sich in zwei Halbbyte (Nibble). Das höherwertige Nibble hält die Information über die Art des Programms, welches gestartet werden soll, das niederwertige den Mode, in dem der Start vollzogen werden soll. (Liste der Bedeutungen siehe DW137).

Hinweis!

Die Information wird im Allgemeinen dem Startsignal des HMI's entnommen. In besonderen Anwendungsfällen (direkter Start durch SPS) erzeugt die SPS die Information selbst.

Wirkung im NC: Der Wert "1" des Signals bewirkt ein Auslösen des definierten Programms im entsprechenden Modus.

Eventuelle Eintragungen in ein P-Feld müssen vorher vorgenommen werden (siehe DW137). Das Signal "NC-Programm läuft" des entsprechenden Kanals wird auf "1" gesetzt.

Datenwort	Name	Richtung	Signalart
009.08-11	Freigabe Override	SPS → NC	statisch

Steuert die Wirksamkeit der Override-Tasten (SPS oder HMI).

Wirkung im NC: Beim Wert "1" werden die von der SPS im Datenbaustein 1 und Datenwort 80-83 gesetzten Overrides, ansonsten die über "virtuelle Tastatur" gesetzten MMI-Overrides berücksichtigt.

Datenwort	Name	Richtung	Signalart
012.00	Programmstop	SPS → NC	statisch

Das "Stop"-Signal wird auf den Wert "1" gesetzt, wenn das Stop-Signal vom HMI an die SPS (DW142) den Wert "1" hat oder ein entsprechender Eingang der SPS gesetzt ist ("Betätigen der Stop-Taste") oder aus anderen Gründen die Voraussetzungen für die weitere Bearbeitung eines Programms entfallen.

Der Wert wird auf "0" gesetzt, wenn das Signal "NC-Programm läuft" bzw. "NC-Programm läuft 1" im Zustand "Unterbrechen" den Wert "0" annimmt.

Wirkung im NC: Der Wert "1" hat eine von der eingestellten Betriebsart abhängige Wirkung im NC.

Handbetrieb: Die laufende Bewegung wird abgebrochen, eventuell anstehende weitere Aufgaben werden verworfen. Das Signal "NC-Programm läuft" wird zu "0" gesetzt.

Automatik: Abbruch des laufenden NC-Programms. Der NC nimmt den Ausgangszustand an. Das Signal "NC-Programm läuft" wird zu "0" gesetzt.

Datenwort	Name	Richtung	Signalart
012.08	Unterbrechen	SPS → NC	statisch

Mit der pos. Flanke des Unterbrechen-Signals kann ein laufendes Programm oder ein durch einen Fehler der Klasse 3 angehaltenes Programm unterbrochen werden (Anmerkung: Ein Programm wird bei einem Fehler der Klasse 3 angehalten, wenn die Maschinenkonstante MK_FEHLERRESTART auf "1" gesetzt ist. Ansonsten wird das Programm abgebrochen.). In diesem Zustand können die Achsen von Hand gefahren und Einzelfunktionen oder andere Programme ausgeführt werden.

Der Unterbrechen-Zustand kann, falls kein Programm und keine Einzelfunktion mehr aktiv ist, durch eine pos. Flanke des Unterbrechen-Signals beendet werden. Das unterbrochene Programm kann danach mit einem Normalstart

wieder aufgenommen oder mit der Stoptaste abgebrochen werden. Bei der Wiederaufnahme erfolgt zunächst eine Ausgleichsfahrt der Achsen auf kürzestem Weg zum Unterbrechenspunkt mit 1/10 der max. Verfahrgeschwindigkeit.

Das Unterbrechen-Signal wird verodert mit dem Unterbrechen-Signal in der virtuellen Tastatur.

Wirkung im NC: NC wechselt mit der 1. pos. Flanke in den Zustand Unterbrechen. Mit der nächsten pos. Flanke wird der Unterbrechen-Zustand wieder verlassen, wenn keine anderen Aktionen aktiv sind.

Hinweis!

Im Unterbrechen-Zustand können keine Koordinatenverschiebungen oder Nullpunktkorrekturen durchgeführt werden.

Datenwort	Name	Richtung	Signalart
013.00 - 07	Einzel-/Folgesatz	SPS → NC	statisch

Umschalten zwischen Folge- und Einzelsatzbetrieb. (Dieses Signal verodert mit dem Signal ”Einzel-/Folgesatz” aus dem DB15.)

Wirkung im NC: Bearbeitung eines Programms kontinuierlich (Folgesatz) oder satzweise (Einzelsatz):

0 = Folgesatz
1 = Einzelsatz

Datenwort	Name	Richtung	Signalart
013.08 - 15	Satz ausblenden	SPS → NC	statisch

Mit diesem Signal kann die Funktion ”Satz ausblenden” ein- oder ausgeschaltet werden. (Dieses Signal wird verodert mit dem Signal ”Satz ausblenden” aus dem DB15).

Wirkung im NC: Sätze mit einem vorangestellten ”/” werden ausgeblendet, d. h. sie werden nicht ausgeführt:

0 = Satz ausblenden inaktiv
1 = Satz ausblenden aktiv

Datenwort	Name	Richtung	Signalart
014.00 - 07	Rückzug	SPS → NC	statisch

Das Signal (ein Bit pro NC-Kanal) ermöglicht den Rückzug (”Zurückfahren”) auf der programmierten Bahn. Die Rückzugsbewegung wird dabei nicht durch Sätze ohne Geometrie eingeschränkt, d. h. einzelne S/T/M/Q-Funktionen, Koordinatenverschiebungen und G21-Sätze sind zugelassen und führen nicht zur Begrenzung des Rückzugsspeichers.

Alle anderen G-Funktionen, die keine reinen Interpreterfunktionen (📖 90) sind, führen wie bisher zur Begrenzung des Rückzugsspeichers. M-Funktionen werden beim Rückzug wie programmiert ausgegeben, wenn dafür keine andere kundenspezifische Lösung vorgesehen ist. G21 und Q-Funktionen werden beim Rückzug ignoriert.

Wirkung im NC: Wenn das Signal gesetzt ist, arbeitet der NC die Sätze, die im Vorlaufpuffer stehen rückwärts ab, wobei Verfahrsätze mit negativer Geschwindigkeit ausgegeben werden.

Datenwort	Name	Richtung	Signalart
018.00	Q-Bit-Signal 0	SPS → NC	statisch
...	...	...	...
021.15	Q-Bit-Signal 63	SPS → NC	statisch

Applikations- und programmspezifische Signale zur Steuerung des Programmablaufs in der NC-Steuerung. Die Bedeutung der Werte ist im Einzelnen zu definieren. Die Werte ergeben sich aus Eingängen der SPS und internen Verknüpfungen.

Wirkung im NC: Die Signale werden unverändert im internen "Q-Feld" als Q0 ... Q63 abgelegt. Auf die einzelnen Einträge des Q-Feldes kann im Programm zugegriffen werden, so dass eine Steuerung des Programmablaufs möglich ist.

Schnelle Eingänge: Falls schnelle Eingänge konfiguriert sind (s. DW36.0 ... 37.15), werden die zugehörigen Q-Bits nicht aus dem Q-Feld sondern direkt von den entsprechenden Eingängen E0.0 ... E3.7 gelesen. Dadurch kann auf ein externes Ereignis schneller reagiert werden, da der Umweg über die integrierte SPS entfällt.

Datenwort	Name	Richtung	Signalart
031.00 - 15	Freigabe MMI	SPS → NC	statisch

Dieses Datenwort wird bisher nur im Zusammenhang mit dem Standard-HMI verwendet. Es wird benutzt, um dem MMI zu signalisieren, welche SPS Funktionen durch die geladene SPS unterstützt werden (bitweise codiert).

Bitnummer	Bedeutung (SPS Unterstützung)
0	Teachen
1	Satzvorlauf
2	Unterbrechen
3	Versionsausgabe
4	Tastenkontrolle

Durch das Setzen der Bits 0 ... 2 wird am HMI ein entsprechender Softkey eingeblendet. Ist Bit 3 gesetzt, fordert das HMI über die Nachrichtenschnittstelle im DB2 einen Versionsstring an. Ist Bit 4 gesetzt, hat die SPS die Tastenkontrolle über die im ETC-MMI angebotenen SPS-Tasten (🕮 262).

Wirkung im NC: keine

Datenwort	Name	Richtung	Signalart
032.00	Quittung für M-Funktion	SPS → NC	Quittung

Das Signal wird auf ”0” gesetzt, wenn die SPS den Wert ”0” des Strobes für M-Funktionen erkannt hat und die von der NC übermittelten Daten der M-Funktionen übernommen hat. Es wird auf ”1” gesetzt, wenn der Strobe wieder den Wert ”1” angenommen hat (📖 237).

Wirkung im NC: Wenn das Signal den Wert ”1” hat, dürfen neue Daten an die SPS übermittelt werden, bei Wert ”0” hat die SPS die anliegenden Daten übernommen. In diesem Fall wird der Strobe von ”0” auf ”1” gesetzt zur Signalisierung, dass die Übernahme erkannt wurde. Beim Wechsel des Signals von ”1” auf ”0” beginnt der NC mit der Bearbeitung des nächsten Satzes, sofern die ”Einlesefreigabe” erteilt ist.

Datenwort	Name	Richtung	Signalart
033.00	Quittung für H-Funktion	SPS → NC	Quittung

Das Signal wird auf ”0” gesetzt, wenn die SPS den Wert ”0” des Strobes für H-Funktionen erkannt und die von der NC übermittelten Daten der H-Funktionen übernommen hat. Es wird auf ”1” gesetzt, wenn der Strobe wieder den Wert ”1” angenommen hat.

Wirkung im NC: Wenn das Signal den Wert ”1” hat, dürfen neue Daten an die SPS übermittelt werden, bei Wert ”0” hat die SPS die anliegenden Daten übernommen. In diesem Fall wird der Strobe vom NC von ”0” auf ”1” gesetzt (Signalisierung Übernahme erkannt). Bei länger andauernder Datenübernahme (>10 sec.) muss das Signal ”Einlesefreigabe” weggenommen werden. Dies muss vor dem Setzen des Quittungssignals erfolgen.

Beim Wechsel des Signals von ”1” auf ”0” beginnt der NC mit der Bearbeitung des nächsten Satzes, sofern die ”Einlesefreigabe” erteilt ist.

Datenwort	Name	Richtung	Signalart
035.00	Freigabe ext. Synchronisationssignal	SPS → NC	Freigabe

Bedeutung: reserviert

Wirkung im NC: reserviert

Datenwort	Name	Richtung	Signalart
035.08	Programmhalt aktiv	SPS → NC	statisch

Durch eine ”1” signalisiert die SPS dem HMI, dass die Abarbeitung eines NC-Programms angehalten wurde.

Wirkung im NC: Keine

Datenwort	Name	Richtung	Signalart
036.00-037.15	Freigabe der ”schnellen Eingänge”	SPS → NC	statisch

Durch eine ”1” deklariert die SPS den entsprechenden digitalen Eingang als ”schnellen Eingang”, d. h. das Eingangssignal wird direkt vom Grobinterpolator (im Grobinterpolationstakt) ausgewertet. Wird die Freigabe durch die SPS wieder zurückgesetzt, stehen ihr ab sofort die Eingänge wieder exklusiv zur Verfügung.

Hinweis!

Der Zustand des ”schnellen Eingangs” ergibt sich aus einer ODER-Verknüpfung des Q-Bit-Signals der SPS und dem Zustand des digitalen Eingangs.

Durch die Datenworte ”Offset der schnellen Eingänge” kann die Freigabe (Maske) als ”schneller Eingang” in jeden Bereich des Eingangsabbildes gelegt werden. Damit kann jeder digitale Eingang als ”schneller Eingang” deklariert werden.

Wirkung im NC: Durch entsprechende DIN-Programmierung (G150, G151) ist es möglich, zeitkritische digitale Eingänge direkt unter Umgehung der SPS durch den Grobinterpolator als Q-Bits auswerten zu lassen.

Beispiel: Mit den folgenden Zeilen werden die Eingänge %IX64.1, %IX64.4, %IX64.13 und %IX65.7 als schnelle Eingänge konfiguriert.

```
DB1_SPS2NC_QIN_OFFSET_AB[0] := 64;
DB1_SPS2NC_QIN_OFFSET_AB[1] := 65;
DB1_SPS2NC_QIN_MASK_AW[0] := 2#0010000000010010;
DB1_SPS2NC_QIN_MASK_AW[1] := 2#0000000010000000;
```

Datenwort	Name	Richtung	Signalart
038.00-039.15	Freigabe der ”schnellen Ausgänge”	SPS → NC	statisch

Eine ”1” entzieht den entsprechenden Ausgang der Kontrolle der SPS und weist diesen dem Grobinterpolator zu; d. h. ein als ”schneller Ausgang” freigegebener Ausgang, kann ausschließlich vom Grobinterpolator bedient werden. Sobald die SPS die Freigabe wieder wegnimmt, stehen ihr die Ausgänge wieder selbst zur Verfügung; die Ausgangssignalzustände werden sofort auf die Ausgänge reflektiert.

Durch die Datenworte ”Offset der schnellen Ausgänge” kann die Freigabe (Maske) als ”schneller Ausgang” in jeden Bereich des Ausgangsabbildes gelegt werden. Damit kann jeder digitale Ausgang als ”schneller Ausgang” deklariert werden.

Nach Übernahme der Kontrolle bleiben die letzten Zustände erhalten.

Wirkung im NC: Durch entsprechende DIN-Programmierung (Parameter Q), ist es möglich, zeitkritische digitale Ausgänge direkt durch den Grobinterpolator (im Grobinterpolationstakt) bedienen zu lassen.

Datenwort	Name	Richtung	Signalart
042.00-043.15	Offset der ”schnellen Eingänge”	SPS → NC	statisch

Mit den beiden Datenworten können die 2 Masken ”Freigabe der schnellen Eingänge” an eine beliebige Stelle im Prozessabbild gelegt werden.

Wirkung im NC: Die entsprechenden Eingänge werden dem NC als Q-Bits zur Verfügung gestellt.

Beispiel: siehe ”Freigabe der schnellen Eingänge”

Datenwort	Name	Richtung	Signalart
044.00-045.15	Offset der ”schnellen Ausgänge”	SPS → NC	statisch

Mit den beiden Datenworten können die 2 Masken ”Freigabe der schnellen Ausgänge” an eine beliebige Stelle im Prozessabbild gelegt werden. Standardmäßig wirkt die Freigabe der ”schnellen Ausgänge” auf die 1. digitale E-/A-Karte (EC-IO).

Wirkung im NC: Die Q-Bits werden an die entsprechenden Ausgänge weitergegeben.

Datenwort	Name	Richtung	Signalart
080	Allgemeiner Override für Achsen	SPS → NC	statisch
081	Override für Spindeldrehzahl	SPS → NC	statisch
082	Override für Oszillationsgeschwindigkeit	SPS → NC	statisch
083	Override für SPS-Achsen (Zielpunktfahrt)	SPS → NC	statisch

Bestimmung eines vorzeichenbehafteten Bewertungsfaktors für die momentan gültige Verfahrgeschwindigkeit. Die Angabe erfolgt in 0,1 % Schritten.

Wirkung im NC: Der ausgewählte Wert wird als Faktor zur Bewertung der momentan programmierten Maximalgeschwindigkeit vom Grobinterpolator verwendet.

Der Override ist sowohl im Handbetrieb als auch Automatik wirksam. Er kann durch M-Funktionen innerhalb eines Programms gesperrt werden (M48/M49 Override ein/aus).

Datenwort	Name	Richtung	Signalart
084.00-07	Verfahrtaste Achse 0	SPS → NC	statisch
084.08-15	Verfahrtaste Achse 1	SPS → NC	statisch
...	...	...	...
...	...	...	....
091.00-07	Verfahrtaste Achse 14	SPS → NC	statisch
091.08-15	Verfahrtaste Achse 15	SPS → NC	statisch

Diese Signale dienen zum Verfahren der Achsen. Für jede Achse steht ein Byte zur Verfügung. Durch Beschreiben der Bytes wird die jeweilige Aktion ausgeführt. Diese Verfahrtasten sind nur aktiv, wenn die Verfahrtastenfreigabe nicht gesetzt ist.

Wirkung im NC:

-100 ... +100: Verfahren mit % der Maximalgeschwindigkeit. Das Vorzeichen gibt die Richtung an.

+/-101: Tippen um Schrittweite (entsprechend DB15 DW44 ... 47) auf der Bahn bei fehlender Vorschubfreigabe. Nur bei einer programmierten Verfahrbewegung verfügbar.

+/-102: Referenzpunktfahrt (Vorzeichen bleibt unbeachtet)

+/-103: Nullpunktsetzen (Vorzeichen bleibt unbeachtet)

+/-104: Schrittfahren (P673 Schrittweite)

+/-105: Handrad aktiv - Bewertungsfaktor 1

+/-106: Handrad aktiv - Bewertungsfaktor 3

+/-107: Handrad aktiv - Bewertungsfaktor 10

+/-108: Handrad aktiv - Bewertungsfaktor 30

+/-109: Handrad aktiv - Bewertungsfaktor 100

+/-110: Verfahrbefehl für SPS-Achse

+/-111: Verfahren mit der Geschwindigkeit aus P208-223.

Das Vorzeichen gibt die Richtung an.

Beschreibung des Verfahrbefehls für SPS-Achsen: Die SPS kann eine beliebige Achse auf eine bestimmte Position fahren, die im P-Feld eingetragen wurde. Dazu werden folgende Parameter verwendet:

P176 ... 191 = Zielpositionsvorgabe von SPS für Achsen 0 ... 15 [mm] bzw. [°]

P208 ... 223 = Geschwindigkeitsvorgabe von SPS für Achsen 0 ... 15 [mm/min] bzw. [U/min]

P240 ... 255 = Beschleunigungsvorgabe von SPS für Achsen 0 ... 15 [m/sec^2] bzw. [U/sec^2]

Zur Anzeige stehen folgende Parameter zur Verfügung:

P144 ... 159 = übernommene Zielposition nach positiver Flanke der Verfahrtaste [mm] bzw.[°]

P160 ... 175 = aktuelle modale Ist-Position [mm] bzw. [°]

Eine Zielpunktfahrt einer Achse wird gestartet, wenn in ihre Verfahrtaste der Wert +/-110 geschrieben wird. Dabei ist darauf zu achten, dass der NC-Rechner eine Änderung der Verfahrtaste erkennt (evtl. Handshake beachten; siehe Diagramm). Genau dann wird der Zielpunkt übernommen. (Anmerkung: Auch eine Änderung von 110 nach –110 oder umgekehrt bewirkt eine erneute Übernahme der Zielposition.) Die Achse fährt, solange der Wert +/-110 in der Verfahrtaste ansteht und das Ziel noch nicht erreicht ist. Eine Änderung der Geschwindigkeitsvorgabe (P208 ... 223) wird auch während der Fahrt übernommen. Das Erreichen des Zielpunktes signalisiert der NC im DB1 mit dem Signal ”SPS Achse am Ziel”.

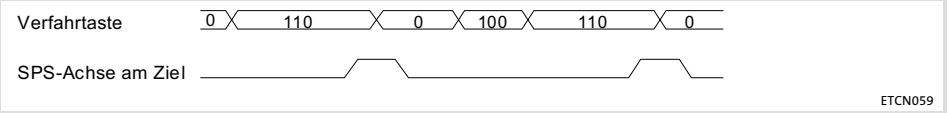
Gefahren wird generell über den ”modalen Offset”, d. h. parallel zu einer eventuell programmierten NC-Zielposition. Bei Übernahme der Zielpositionsvorgabe aus dem P-Feld werden je nach Konfiguration der Verfahrtaste zwei Fälle unterschieden:

1. G122 X0 (Default-Konfiguration)

Die modale Zielposition ergibt sich aus der Differenz Zielpositionsvorgabe von SPS und aktueller NC-Ist-Position. Die Vorgabe wirkt also relativ zum aktuellen Koordinatensystem.

2. G122 X1

Die modale Zielposition ergibt sich direkt aus der Zielpositionsvorgabe von der SPS. Die Vorgabe wirkt also relativ zur aktuellen NC-Ist-Position.



Datenwort	Name	Richtung	Signalart
128.00	NC-Betriebsbereit_1	NC → SPS	statisch

Der Wert ”1” zeigt die Betriebsbereitschaft der NC-Steuerung in allen Betriebsarten an. Das ”0”-Signal erscheint, wenn die NC-Steuerung in einen behebbaren Fehlerzustand wechselt, der zu einer Unterbrechung des Programmablaufs führt (Signal ”NC-Programm läuft” bleibt weiter auf ”1”).

Das Signal wechselt von ”0” auf ”1”, wenn die Fehlerursache behoben wird (Fehlerquittierung durch Bediener).

Wirkung in SPS: nicht festgelegt

Datenwort	Name	Richtung	Signalart
128.01	NC-Betriebsbereit_2	NC → SPS	statisch

Der Wert ”1” zeigt die Betriebsbereitschaft der NC-Steuerung in allen Betriebsarten an. Das ”0”-Signal erscheint, wenn die Steuerung in einen nicht behebbaren Fehlerzustand wechselt, der zu einem Abbruch des Programmablaufs führt oder im ”NOT AUS” Zustand ist. Das Signal wechselt von ”0” auf ”1”, wenn die Fehlerursache behoben wird (Fehlerquittierung durch Bediener) und/oder die Steuerung nicht im ”NOT-AUS” Zustand ist.

Wirkung in SPS: nicht festgelegt

Beispiel Wechsel in NOT-AUS Zustand, Wegnahme der Reglerfreigabe

Datenwort	Name	Richtung	Signalart
129.00-15	Referenzpunkt Achse 0 – 15	NC → SPS	statisch

Das Signal hat den Wert ”0” nach dem Einschalten der Steuerung und unbekanntem Referenzpunkt. Das Signal hat den Wert ”1”, wenn der Referenzpunkt der jeweiligen Achse angefahren wurde oder wenn bei dem Wert für die Maschinenkonstante MK_REF_RICHTUNG_UND_FOLGE für die jeweilige Achse -1 eingetragen ist.

Wirkung in SPS: Überprüfung der Referenzpunktfahrt und Verriegelung von Maschinenfunktionen oder Verfahrbewegungen (Vorschubfreigabe) bei nicht angefahrenem Referenzpunkt.

Datenwort	Name	Richtung	Signalart
130.00-15	Referenzfahrt läuft	NC → SPS	statisch

Das Signal hat den Wert ”1”, wenn die Referenzfahrt der jeweiligen Achse gerade läuft.

Wirkung in SPS: Überprüfung, ob die Referenzpunktfahrt einer Achse gerade aktiv ist.

Datenwort	Name	Richtung	Signalart
131	Betriebsart	NC → SPS	statisch

Das Betriebsartenwort zeigt der SPS die vom Bediener gewählte Betriebsart an, in der sich die NC-Steuerung augenblicklich befindet:

0 = Grundzustand; Keine Achsenbewegung möglich
1 = Einrichten
2 = Einrichtfunktionen: modales Fahren, Schrittfahren, Nullpunkte
3 = Automatik, autom. Bearbeitung von Programmen (Einzelsatz, Folgesatz)
4 = Programmieren
80H = Diagnose (DW131.7)

Die Betriebsarten werden vom Bediener vorgewählt. Bei Verwendung des Lenze-MMI wird die gewählte Unterbetriebsart im DB2 in DW128 abgelegt.

Wirkung in SPS: Betriebsartspezifische Freigaben bzw. Verriegelungen.

Datenwort	Name	Richtung	Signalart
132.00	NC-Programm läuft	NC → SPS	Statisch

Der "1"-Wert des Signals zeigt an, dass die Bearbeitung eines NC-Programms oder eines Verfahrsatzes ("Hand Einzelsatz") aktiv ist. Das Signal bleibt anstehen bis zum Programmende (M02, M30), Programmabbruch ("NC-Programmstop") oder internem Abbruch durch nicht behebbare Fehler bzw. dem Satzende oder dem Abbruch des Satzes (Handbetrieb).

Wirkung in SPS: Auswertung für Verriegelungen, Synchronisation, Zeitmessungen.

Datenwort	Name	Richtung	Signalart
132.02	Satzvorlauf aktiv	NC → SPS	Statisch

Der "1"-Wert des Signals zeigt an, dass die Bearbeitung eines NC-Programms im Satzvorlauf aktiv ist. Das Signal bleibt anstehen bis der Zielsatz des Satzvorlaufs erreicht ist.

Datenwort	Name	Richtung	Signalart
132.03	Einzelsatz läuft	NC → SPS	Statisch

Der "1"-Wert des Signals zeigt an, dass die Bearbeitung eines Einzelsatzes, ausgelöst durch Einzelsatz-Funktion (s. ETC MMI) oder durch Eingabe am MMI (Handbetrieb), aktiv ist. Das Signal bleibt anstehen bis zum Satzabbruch ("NC-Programmstop") oder internem Abbruch durch nicht behebbaren Fehler.

Wirkung in SPS: Auswertung für Verriegelungen, Synchronisation, Zeitmessungen.

Datenwort	Name	Richtung	Signalart
0132.04	Unterbrechen aktiv	SPS → NC	statisch

Der "1"-Wert des Signals zeigt an, dass der NC im Zustand "Unterbrechen" ist (s. a. Signal "Unterbrechen").

Wirkung in SPS: Keine.

Datenwort	Name	Richtung	Signalart
132.08	Alle Achsen stehen	NC → SPS	statisch

Das Signal hat den Wert ”1”, immer wenn alle Achsen stehen bzw. kein Auftrag zum Verfahren der Achse aktiv ist.

Das Signal ”Alle Achsen stehen” wird aus den Zustandssignalen aller Achsen gebildet.

Datenwort	Name	Richtung	Signalart
133.00-07	Wahlweise Halt	NC → SPS	statisch

Durch dieses Signal vom MMI wird ein ”Programmierter Halt” durch M01 aktiviert werden.

Wirkung im NC:

0 = wahlweise Halt inaktiv
1 = wahlweise Halt aktiv

Datenwort	Name	Richtung	Signalart
133.08-15	Programmstartzähler	NC → SPS	statisch

Dieser Zähler wird bei jeder steigenden Flanke von ”NC-Programm läuft” um 1 inkrementiert.

Datenwort	Name	Richtung	Signalart
134.00-15	Achse fährt, Achse 0 ... 15	NC → SPS	statisch

Das Signal hat den Wert ”1”, immer wenn eine Achse in Bewegung ist bzw. ein Auftrag zum Verfahren der Achse aktiv ist. Das Signal hat also auch dann den Wert ”1”, wenn die Bewegung der Achse momentan unterbrochen ist durch ”Vorschubfreigabe” oder ”override=0”. Das Signal wird bei Erreichen des Ziels auf ”0” gesetzt.

Wirkung in SPS: Spezifische Verriegelungen (z. B. im Handbetrieb) bei sich bewegenden Achsen.

Datenwort	Name	Richtung	Signalart
135.00-15	SPS-Achse am Ziel, Achse 0 -... 15	NC → SPS	statisch

Das Signal hat den Wert ”1”, wenn eine Achse aus der SPS gestartet wurde, das Ziel erreicht hat und der Verfahrbefehl 110 in der Verfahrtaste noch ansteht.

Wirkung in SPS: Die Verfahrtaste kann wieder auf Null gesetzt werden.

Datenwort	Name	Richtung	Signalart
137.00-07	Programmstart	NC → SPS	statisch

Das Startsignal ist eine Byte-Information, die einen Wert zwischen 0 und 255 annehmen kann. Ein Wert größer ”0” wird ausgegeben, wenn die Information im virtuellen Tastenfeld des NC einen Wert ungleich ”0” aufweist. Das Signal liegt an, solange die Information im virtuellen Tastenfeld ansteht. Die Information gliedert sich in zwei Halbbytes (Nibble). Das höherwertige Nibble hält die Information über die Art des Programms, welches gestartet werden soll, das niederwertige den Mode, in dem der Start vollzogen werden soll. Die Information wird durch das MMI bestimmt.

Werte des höherwertigen Nibbles

0	Start des im P-Feld (512) eingetragenen NC-Programms
1	Start des im P-Feld (514) eingetragenen NC-Programms
2	Start des im P-Feld (516) eingetragenen NC-Programms
3	Start des im P-Feld (518) eingetragenen NC-Programms
4	Start des im P-Feld (520) eingetragenen NC-Programms
5	Start des im P-Feld (522) eingetragenen NC-Programms
6	Start des im P-Feld (524) eingetragenen NC-Programms
7	Start des im P-Feld (526) eingetragenen NC-Programms
8	
9	
10	
11	
12	
13	
14	
15	Start der automatischen Referenzpunktfahrt

Werte des niederwertigen Nibbles

0	keine Reaktion
1	Start (normale Ausführung)
2	Start Satzvorlauf (P528-P532 Konfiguration des Einstiegspunktes)
3	
4	
5	
6	
7	
8	
9	
10	
11	Wiederaufnahme eines durch Fehler unterbrochenen Programms an einer definierten Stelle (G10)
12	Wiederaufnahme eines unterbrochenen Programms nach vorheriger (implicit) Rückkehr in Normalbetrieb
13	Wiederaufnahme eines durch Fehler unterbrochenen Programms ohne vorherige Korrekturmaßnahme
14	Start nach M0/M1
15	Start des nächsten Satzes im Einzelbetrieb

Wirkung in SPS: Überprüfung der Startvoraussetzungen für den jeweiligen Startmode und gegebenenfalls Freigabe des Programmstarts durch Setzen des Signals ”Programmstart” (DW 009). Absetzen einer entsprechenden Fehlermeldung, wenn der Start verweigert wird.

Datenwort	Name	Richtung	Signalart
137.08 - 15	Einzel-/ Folgesatz aktiv	NC → SPS	statisch

Umschalten von Folge- auf Einzelsatzbetrieb und umgekehrt. (Kopie aus der virtuellen Tastatur).

Wirkung im NC: Bearbeitung eines Programms kontinuierlich (Folgesatz) oder satzweise (Einzelsatz).

0 = Folgesatz; 1 = Einzelsatz

Datenwort	Name	Richtung	Signalart
138.00-15 139.00-15	Status CAN-Module (SLIO)	NC → SPS	statisch

Datenwort 138.00-15: Signalzustand ”1” signalisiert, dass das entsprechende CAN-Modul vorhanden ist.

Datenwort 139: Ein Wert ungleich 0 bedeutet, dass ein Fehler im Zusammenhang mit den CAN-Modulen aufgetreten ist.

Wirkung in SPS: Möglichkeit der Überwachung der CAN-Module.

Datenwort	Name	Richtung	Signalart
142.00	Programmstop	NC → SPS	statisch

Abbild der ”Stop” Taste. Das Signal hat den Wert ”1”, solange die Information in der virtuellen Tastatur des NC den Wert ”1” hat. Mit dem Wert ”1” wird der Abbruch eines laufenden Programms bzw. einer Verfahrbewegung angefordert. Die Bedeutung ist abhängig von der eingestellten Betriebsart.

Handbetrieb: Abbruch der laufenden Bewegung

Automatik: Abbruch des laufenden Programms

Wirkung in SPS: Bei Erkennung des Wertes ”1” wird nach der Überprüfung evtl. vorhandener weiterer Bedingungen das Signal ”Programmstop” (DW 012.XX) gesetzt.

Datenwort	Name	Richtung	Signalart
144.00	Vorschub Halt	NC → SPS	statisch

Abbild der Tasten ”Vorschub Halt” bzw. ”Feedhold”. Es hat den Wert ”1”, solange die Information in der virtuellen Tastatur des NC den Wert ”1” hat. Mit dem Wert ”1” wird ein Stehen bleiben der Achsen ohne Programmabbruch oder Abbruch einer Verfahrbewegung angefordert.

Wirkung in SPS: Bei Erkennung des Wertes ”1” des Signals und Überprüfung gegebenenfalls vorhandener weiterer Bedingungen nimmt die SPS das Signal ”Vorschubfreigabe” für alle Achsen weg. Die Vorschubfreigabe muss wiedergegeben werden bei ”Programm Start” und ”Start nach Vorschub Halt”.

Datenwort	Name	Richtung	Signalart
150.00	Strobe M-Funktion	NC → SPS	Strobe

Das Signal wird auf ”0” gesetzt, wenn das Datenwort für die ”M-Funktion” gültig ist. Es wird auf den Wert ”1” gesetzt, wenn das Quittungssignal der SPS um ”1” auf ”0” gewechselt hat. Nach dem Einschalten der Steuerung hat das Signal den Wert ”1”.

Wirkung in SPS: Wenn das Signal den Wert ”0” hat, wird das Datenwort der ”M-Funktion” übernommen. Das Quittierungssignal wird nach der Übernahme auf den Wert ”0” gesetzt. Bei Wert ”1” des Strobe wird das Quittungssignal auf ”1” gesetzt.

Bei synchronisierten M-Funktionen wird das Signals ”Einlesefreigabe” weggenommen. Dies erfolgt vor dem Setzen des Quittungssignals.

Datenwort	Name	Richtung	Signalart
151	M-Funktion	NC → SPS	Nachricht

Das Datenwort enthält die Nummer der M-Funktion in binärer Darstellung (0 ... 9999). Die M-Funktion kann im Teileprogramm programmiert sein oder durch Tastendruck erzeugt werden.

Die Nummern der M-Funktionen sind teilweise durch Normen festgelegt (DIN 66025), die freien Nummern können anwendungsspezifisch vergeben werden. Eine Liste der für die ETC festgelegten M-Nummern und deren Beschreibung findet sich in der ETC Programmierung NC-Betriebssystem.

Wirkung in SPS: Ausführung der durch die M-Funktion spezifizierten Funktion. Nach Beendigung der Funktion, oder zu einem geeigneten Zeitpunkt, wird wieder die Einlesefreigabe erteilt.

Datenwort	Name	Richtung	Signalart
159.00	Strobe H-Funktion	NC → SPS	Strobe

Das Signal wird auf ”0” gesetzt, wenn das Datenwort für die ”H-Funktion” gültig ist. Es wird auf den Wert ”1” gesetzt, wenn das Quittungssignal der SPS von ”1” auf ”0” gewechselt hat. Nach dem Einschalten der Steuerung hat das Signal den Wert ”1”.

Wirkung in SPS: Wenn das Signal den Wert ”0” hat, wird das Datenwort der ”H-Funktion” übernommen. Das Quittierungssignal wird nach der Übernahme auf den Wert ”0” gesetzt. Bei Wert ”1” des Strobe wird das Quittungssignal auf ”1” gesetzt.

Bei länger andauernder Datenübernahme (>10 sec.) muss das Signal ”Einlesefreigabe” weggenommen werden. Dies muss vor dem Setzen des Quittungssignals erfolgen.

Datenwort	Name	Richtung	Signalart
160	H-Funktion	NC → SPS	Nachricht

Das Datenwort enthält die Nummer der H-Funktion in binärer Darstellung. Die Beschreibung zur Übernahme einer H-Funktion befindet sich in der ”Programmieranleitung”.

Wirkung in SPS: Ausführung der durch die H-Funktion spezifizierten Funktion. Nach Beendigung der Funktion oder zu einem geeigneten Zeitpunkt wird wieder die Einlesefreigabe erteilt

Datenwort	Name	Richtung	Signalart
177.00-15	Referenzpunkt angefahren	NC → SPS	statisch

Das Signal hat den Wert ”1”, wenn eine Referenzpunktfahrt der jeweiligen Achse erfolgreich durchgeführt wurde, den Wert ”0” nach dem Einschalten der Steuerung und unbekanntem Referenzpunkt.

Wirkung in SPS: Überprüfung der Referenzpunktfahrt und Verriegelung von Maschinenfunktionen oder Verfahrbewegungen (Vorschubfreigabe) bei nicht angefahrenem Referenzpunkt.

Datenwort	Name	Richtung	Signalart
178.00-15	Endschalter + aktiv	NC → SPS	statisch

Das Signal hat den Wert ”1”, wenn der positive Endschalter einer Achse aktiv ist, sonst den Wert ”0”.

Wirkung in SPS: Verriegelung von Maschinenfunktionen oder Verfahrbewegungen (Vorschubfreigabe).

Datenwort	Name	Richtung	Signalart
179.00-15	Endschalter - aktiv	NC → SPS	statisch

Das Signal hat den Wert ”1”, wenn der negative Endschalter einer Achse aktiv ist, sonst den Wert ”0”.

Wirkung in SPS: Verriegelung von Maschinenfunktionen oder Verfahrbewegungen (Vorschubfreigabe).

Datenwort	Name	Richtung	Signalart
180.00-15	Referenzschalter aktiv	NC → SPS	statisch

Das Signal hat den Wert ”1”, wenn der Referenzschalter einer Achse aktiv ist, sonst den Wert ”0”.

Wirkung in SPS: Evtl. man. Ausführung einer Referenzfahrt durch die SPS.

Datenwort	Name	Richtung	Signalart
182.00-15	Slave-Achse Endschalter + aktiv	NC → SPS	statisch

Das Signal hat den Wert ”1”, wenn der positive Endschalter der Slave-Achse einer Synchronachse aktiv ist, sonst den Wert ”0”.

Wirkung in SPS: Verriegelung von Maschinenfunktionen oder Verfahrbewegungen (Vorschubfreigabe).

Datenwort	Name	Richtung	Signalart
183.00-15	Slave-Achse Endschalter - aktiv	NC → SPS	statisch

Das Signal hat den Wert ”1”, wenn der negative Endschalter der Slaveachse einer Synchronachse aktiv ist, sonst den Wert ”0”.

Wirkung in SPS: Verriegelung von Maschinenfunktionen oder Verfahrbewegungen (Vorschubfreigabe).

Datenwort	Name	Richtung	Signalart
184.00-15	Slave-Achse Referenzschalter aktiv	NC → SPS	statisch

Das Signal hat den Wert ”1”, wenn der Referenzschalter der Slave-Achse einer Synchronachse aktiv ist, sonst den Wert ”0”.

Wirkung in SPS: Eventuelle manuelle Ausführung einer Referenzfahrt durch die SPS.

Datenwort	Name	Richtung	Signalart
192.00-199.15	Status CAN-Module (CANopen)	NC → SPS	statisch

Signalzustand ”1” signalisiert, dass das entsprechende CANopen-Modul vorhanden ist. Die Zuordnung der Bits in den Datenworten erfolgt anhand der Modul-Id (Knotennr.).

Beispiel:

192.01 signalisiert Zustand von Knoten 1
192.15 signalisiert Zustand von Knoten 15
193.01 signalisiert Zustand von Knoten 17

Wirkung in SPS: Möglichkeit der Überwachung der CAN-Module.

Datenwort	Name	Richtung	Signalart
200	Allgemeiner Override für Achsen	NC → SPS	statisch
201	Override für Spindeldrehzahl	NC → SPS	statisch
202	Override für Oszillationsgeschwindigkeit	NC → SPS	statisch
203	Override für SPS-Achsen (Zielpunktfahrt)	NC → SPS	statisch

Bewertungsfaktor, der am HMI eingestellt ist, für die momentan gültige Verfahrgeschwindigkeit. Die Angabe erfolgt in 0,1 % Schritten (Kopie aus der virtuellen Tastatur).

Wirkung im NC: Der ausgewählte Wert wird als Faktor zur Bewertung der momentan programmierten Maximalgeschwindigkeit vom Grobinterpolator verwendet. Der Override ist sowohl im Handbetrieb als auch Automatik wirksam. Er kann z. B. durch M-Funktionen innerhalb eines Programms gesperrt werden.

Datenwort	Name	Richtung	Signalart
204.00-07	Verfahrtaste Achse 0	HMI → SPS	statisch
204.08-15	Verfahrtaste Achse 1	HMI → SPS	statisch
...	...	...	...
...	...	...	....
212.00-07	Verfahrtaste Achse 14	HMI → SPS	statisch
212.08-15	Verfahrtaste Achse 15	HMI → SPS	statisch

Wie Beschreibung für DW 84-91. Nur von Bedeutung, wenn die Verfahrtastenfreigabe der SPS erteilt wird und das MMI über Verfahrtasten verfügt.

Datenwort	Name	Richtung	Signalart
212.00-15 213.00-15 214.00-15 215.00-15 216.00-15 217.00-15 218.00-15 219.00-15	Bereich der ”Virtuellen Tastatur”	HMI → SPS	statisch

Bereich der virtuellen Tastatur dessen Bedeutung/Funktion frei definiert werden kann. Bei Verwendung des ETC-MMI (PC-Bedienoberfläche) werden hier die Zustände der freikonfigurierbaren SPS-Tasten (Softkeys) bitweise abgelegt (siehe Kapitel ”ETC-MMI”).

5.1.3 Datenbaustein 2

Über diesen Datenbaustein wird die Kommunikation zwischen MMI und SPS abgewickelt. Der Datenbaustein wird im Dual-Port-RAM abgebildet.

Datenwort 0 ... 127 enthält Signale von der SPS zum MMI. Datenwort 128 ... 191 enthält Signale vom HMI an die SPS.

Außerdem sind ab Datenwort 224 alle Daten der Maschinenkonstante MK_DW224_255 abgelegt, die applikationsspezifisch genutzt werden können. Dies sind Daten zur Konfiguration der Maschine.

Die Aufteilung der Daten ist dem Anwender überlassen und spezifisch festzulegen. Der NC sorgt lediglich für den Austausch der Daten.

Die Maschinenkonstanten werden beim Laden der Systemsoftware in den Datenbaustein kopiert sowie bei einer zugelassenen Änderung der Maschinenkonstanten. Die Daten werden den allgemeinen Maschinenkonstanten zur Konfiguration der Maschine entnommen. Dort ist ein Bereich reserviert für die Konfiguration der SPS (siehe Kapitel ”Maschinenkonstanten”).

Belegung

Beim ETC-MMI wird die Belegung des DB2 verwendet, die in der folgenden Tabelle aufgeführt ist.

Diese Belegung stellt einen Vorschlag für die Verwendung des DB2 dar. Innerhalb der Bereiche Datenwort 0 ... 127 und 128 ... 191 ist eine frei definierbare Belegung möglich.

Datenwort	Name	Richtung
000.00 – 007.15	Zustände (28 Bits)	SPS → HMI
008.00 – 015.15	Tastenkontrolle (128 Bits)	SPS → HMI
016.00 – 079.15	Anzeigen (64 Datenwort)	SPS → HMI
080.00 – 095.15	Hinweise/statische Fehler (256 Bits)	SPS → HMI
096.00 – 096.15	Nachrichtenpuffer: Quittungszähler	SPS → HMI
097.00 – 097.15	Nachrichtenpuffer: Zähler 1	SPS → HMI
098.00 – 125.15	Nachrichtenpuffer: Nutzdaten (28 Datenworte)	SPS → HMI
126.00 – 126.15	Nachrichtenpuffer: Zähler 2	SPS → HMI
127.00 – 127.15	Überwachung SPS läuft	SPS → HMI
128.00 – 128.15	Unterbetriebsart	SPS ← HMI
129.00 – 143.15	Zustände (256 Bits)	SPS ← HMI
144.00 – 159.15	Tasten (256 Bits)	SPS ← HMI
160.00 – 160.15	Nachrichtenpuffer: Quittungszähler	SPS ← HMI
161.00 – 161.15	Nachrichtenpuffer: Zähler 1	SPS ← HMI
162.00 – 189.15	Nachrichtenpuffer: Nutzdaten (28 Datenwort)	SPS ← HMI
190.00 – 190.15	Nachrichtenpuffer: Zähler 2	SPS ← HMI
191.00 – 191.15	Überwachung HMI läuft	SPS ← HMI
224.00 – 255.15	frei verfügbare Maschinenkonstanten	SPS ← NC

Bei aktivierter Tastenkontrolle (DB1.31.4) korrespondieren die 128 Bits von DB2.8.0 bis DB2.15.15 mit den MMI-Tasten DB1.212.0 bis DB1.219.15.

Erläuterung

DW080.00-095.15	Signalzustand ”1” bewirkt, dass im Fenster SPS-Meldungen am MMI ein Text dargestellt wird. Die Texte sind in einer Datei (default:Lenze.txt) abgelegt, wobei DW080.00 mit dem Eintrag 01020001, DW080.01 mit dem Eintrag 01020002 usw. verknüpft ist.
DW128	Bei Verwendung des ETC-MMI wird hier die gewählte Unterbetriebsart abgelegt: 0: keine Unterbetriebsart gewählt; Betriebsart (DW131) = 1 **Betriebsart (DW131) = 1** 1: Schrittfahren 2: modales Fahren 3: Zielpunktfahren 4: Handrad 5: manuelle Referenzfahrt 6: automatische Referenzfahrt 7: Nullpunkte **Betriebsart (DW131) = 3** 1: Satzvorlauf **Betriebsart (DW131) = 4** 1: Teachen
DW127.00-127.15, DW191.00-191.15	Die Datenworte dienen zur gegenseitigen Kontrolle von SPS und MMI. Damit lässt sich überwachen, ob der jeweilige Gegenpart noch ”lebt”. Funktionsweise: SPS: DW127 := DW191; HMI: DW191 := DW127 + 1; Die SPS kann nun erkennen, wenn sich beide Datenworte über einen bestimmten Zeitraum nicht unterscheiden, dass das MMI nicht mehr ”lebt”.
DW224.00-255.15	Die Werte der Maschinenkonstanten MK_DW224_255 werden eins zu eins in diesen Datenworten abgelegt.

5.2 Erweiterte Schnittstelle für MMI-Funktionen

5.2.1 Datenbausteine 8 ... 14

Belegung

DB8 Datenwort	Name	Richtung
DW000 - 255	reserviert	

DB9 Datenwort	Name	Richtung
DW000 - 255	reserviert	

DB10 Datenwort	Name	Richtung
DW000 - 255	reserviert	

DB11 Datenwort	Name	Richtung
DW000 - 247	reserviert	
DW250 - 253	Zielposition Satz Achse 0	SPS ← NC
DW254 - 255	Zielposition Satz Achse 1 (LOW-WORD)	SPS ← NC

DB12 Datenwort	Name	Richtung
DW000 - 255	reserviert	
DW000 - 001	Zielposition Satz Achse 1 (HIGH-WORD)	SPS ← NC
SW002 - 005	Zielposition Satz Achse 2	SPS ← NC
DW006 - 009	Zielposition Satz Achse 3	SPS ← NC
DW010 - 013	Zielposition Satz Achse 4	SPS ← NC
DW014 - 017	Zielposition Satz Achse 5	SPS ← NC
DW018 - 021	Zielposition Satz Achse 6	SPS ← NC
DW022 - 025	Zielposition Satz Achse 7	SPS ← NC
DW026 - 029	Zielposition Satz Achse 8	SPS ← NC
DW030 - 033	Zielposition Satz Achse 9	SPS ← NC
DW034 - 037	Zielposition Satz Achse 10	SPS ← NC
DW038 - 041	Zielposition Satz Achse 11	SPS ← NC
DW042 - 045	Abstand aktueller Nullpunkt zu Referenzpunkt Achse 0	SPS ← NC
DW046 - 049	Abstand aktueller Nullpunkt zu Referenzpunkt Achse 1	SPS ← NC
DW050 - 053	Abstand aktueller Nullpunkt zu Referenzpunkt Achse 2	SPS ← NC
DW054 - 057	Abstand aktueller Nullpunkt zu Referenzpunkt Achse 3	SPS ← NC
DW058 - 061	Abstand aktueller Nullpunkt zu Referenzpunkt Achse 4	SPS ← NC
DW062 - 065	Abstand aktueller Nullpunkt zu Referenzpunkt Achse 5	SPS ← NC
DW066 - 069	Abstand aktueller Nullpunkt zu Referenzpunkt Achse 6	SPS ← NC
DW070 - 072	Abstand aktueller Nullpunkt zu Referenzpunkt Achse 7	SPS ← NC
DW074 - 077	Abstand aktueller Nullpunkt zu Referenzpunkt Achse 8	SPS ← NC
DW078 - 081	Abstand aktueller Nullpunkt zu Referenzpunkt Achse 9	SPS ← NC
DW082 - 085	Abstand aktueller Nullpunkt zu Referenzpunkt Achse 10	SPS ← NC
DW086 - 089	Abstand aktueller Nullpunkt zu Referenzpunkt Achse 11	SPS ← NC
DW090 - 093	Istposition Satz Achse 0	SPS ← NC
DW094 - 097	Istposition Satz Achse 1	SPS ← NC
DW098 - 101	Istposition Satz Achse 2	SPS ← NC
DW102 - 105	Istposition Satz Achse 3	SPS ← NC
DW106 - 109	Istposition Satz Achse 4	SPS ← NC
DW110 - 113	Istposition Satz Achse 5	SPS ← NC
DW114 - 117	Istposition Satz Achse 6	SPS ← NC

DB12 Datenwort	Name	Richtung
DW118 - 121	Istposition Satz Achse 7	SPS ← NC
DW122 - 125	Istposition Satz Achse 8	SPS ← NC
DW126- 129	Istposition Satz Achse 9	SPS ← NC
DW130 - 133	Istposition Satz Achse 10	SPS ← NC
DW134 - 137	Istposition Satz Achse 11	SPS ← NC
DW137	Schleppabstand Achse 0 in Inkrementen	SPS ← NC
DW139	Schleppabstand Achse 1 in Inkrementen	SPS ← NC
DW140	Schleppabstand Achse 2 in Inkrementen	SPS ← NC
DW141	Schleppabstand Achse 3 in Inkrementen	SPS ← NC
DW142	Schleppabstand Achse 4 in Inkrementen	SPS ← NC
DW143	Schleppabstand Achse 5 in Inkrementen	SPS ← NC
DW144	Schleppabstand Achse 6 in Inkrementen	SPS ← NC
DW145	Schleppabstand Achse 7 in Inkrementen	SPS ← NC
DW146	Schleppabstand Achse 8 in Inkrementen	SPS ← NC
DW147	Schleppabstand Achse 9 in Inkrementen	SPS ← NC
DW148	Schleppabstand Achse 10 in Inkrementen	SPS ← NC
DW149	Schleppabstand Achse 11 in Inkrementen	SPS ← NC
DW150 - 151	Drehzahl Spindel 0	SPS ← NC
DW152 - 153	Drehzahl Spindel 1	SPS ← NC
DW154 - 155	Drehzahl Spindel 2	SPS ← NC
DW156 - 157	Drehzahl Spindel 3	SPS ← NC
DW158 - 159	Drehzahl Spindel 4	SPS ← NC
DW160 - 161	Drehzahl Spindel 5	SPS ← NC
DW174 - 175	Ereigniszähler	SPS ← NC
DW176 - 177	Ereignisabstand	SPS ← NC
DR178	Betriebsart	SPS ← NC
DL178	Programmbearbeitung aktiv	SPS ← NC
DR179	Satzvorlauf	SPS ← NC
DL179	Maßsystem (mm/inch)	SPS ← NC
DR180	Satz ausblenden	SPS ← NC
DL180	Einzelsatz	SPS ← NC
DR181	Wahlweise Halt	SPS ← NC
DL181	Programmierter Halt	SPS ← NC
182.00 - 182.15	Referenzpunkt angefahren ja/nein	SPS ← NC
183.00 - 183.15	Endschalter + angefahren ja/nein	SPS ← NC
184.00 - 184.15	Endschalter – angefahren ja/nein	SPS ← NC
185.00 - 185.15	Referenznocken angefahren ja/nein	SPS ← NC
186.00 - 186.00	Reserve-Eingang (AR)	SPS ← NC
DW194 - 197	Winkel Grunddrehung	SPS ← NC
DW198 - 201	Schleppabstand von Achse 0 in Eingabeeinheiten	SPS ← NC
DW202 - 205	Schleppabstand von Achse 1 n Eingabeeinheiten	SPS ← NC
DW206 - 209	Schleppabstand von Achse 2 in Eingabeeinheiten	SPS ← NC
DW210 - 213	Schleppabstand von Achse 3 in Eingabeeinheiten	SPS ← NC
DW214 - 217	Schleppabstand von Achse 4 in Eingabeeinheiten	SPS ← NC
DW218 - 221	Schleppabstand von Achse 5 in Eingabeeinheiten	SPS ← NC
DW222 - 225	Schleppabstand von Achse 6 in Eingabeeinheiten	SPS ← NC
DW226 - 229	Schleppabstand von Achse 7 in Eingabeeinheiten	SPS ← NC
DW230 - 233	Schleppabstand von Achse 8 in Eingabeeinheiten	SPS ← NC

DB12 Datenwort	Name	Richtung
DW234 - 237	Schleppabstand von Achse 9 in Eingabeeinheiten	SPS ← NC
DW238 - 241	Schleppabstand von Achse 10 in Eingabeeinheiten	SPS ← NC
DW242 - 245	Schleppabstand von Achse 11 in Eingabeeinheiten	SPS ← NC
DW254	Override Achsen	SPS ← NC
DW255	Override Spindeln	SPS ← NC

DW174 - 175: Ereigniszähler; Anzahl der positiven Flanken am "Reserveeingang der ersten Achse".

DW176 - 177: Ereignisabstand; zeitlicher Abstand zwischen zwei Ereignissen (s. DW174 - 175) in Mikrosekunden. Kleinste Auflösung: Feininterpolationstakt.

DB13 Datenwort	Name	Richtung
DR034	reserviert (eigentlich Kanalnummer)	SPS ← NC
DL034	Programm aktiv	SPS ← NC
DR035	aktuelle Werkzeugnummer	SPS ← NC
DL035	aktuelle Werkstücknummer	SPS ← NC
DW036 - 037	Aktivkennung Achse 0 - 3	SPS ← NC
DW038 - 041	Bandgeschwindigkeit	SPS ← NC
DW042	aktuelle Programmnummer	SPS ← NC
DW043	aktuelle programmierte Satznummer	SPS ← NC
DW044	aktuelle relative Satznummer	SPS ← NC
DW045	letzte M-Funktion	SPS ← NC
DW046	aktuelle Verweilzeit	SPS ← NC
DR049	Ebene	SPS ← NC
DL049	Bahn- / Streckenbetrieb	SPS ← NC

DB14 Datenwort	Name	Richtung
DW000 - 255	reserviert	

5.2.2 Datenbaustein 15

Belegung

Der Datenbaustein 15 stellt die ”virtuelle Tastatur” des NC-Rechners dar. Dieser Baustein kann von der SPS nur dann beschrieben werden, wenn kein ETC-MMI an die Steuerung angeschlossen ist. Ansonsten kann der Inhalt von der SPS nur gelesen werden.

Datenwort	Name	Richtung
DR010	Wahlweise Halt	HMI → NC
DL010	Einzel- / Folgesatz	HMI → NC
DR011	Satz ausblenden	HMI → NC
DW028 - 033	Verfahrtasten Achse 0 - 11	HMI → NC
DW036 - 039	Override Wahlschalter 1 - 4	HMI → NC
DW040 - 043	Geschwindigkeitsvorwahl für Verfahrtasten	HMI → NC
DW044 - 047	Schrittweitenvorwahl	HMI → NC

Beschreibung

Datenwort	Name	Richtung	Signalart
010.00-07	Wahlweise Halt	HMI → NC	statisch

Aktivieren der Ausführung von M01. Dieses Signal wird vom NC-Kern nicht ausgewertet, die entsprechende Funktion muss bei Bedarf in der SPS durch Wegnahme der Vorschub- oder Einlesefreigabe bei M01 implementiert werden.

Wirkung im NC:

0 = wahlweise Halt inaktiv
1 = wahlweise Halt aktiv

Datenwort	Name	Richtung	Signalart
010.08-15	Einzel- / Folgesatz	HMI → NC	statisch

Umschalten von Folge- auf Einzelsatzbetrieb und umgekehrt.

Wirkung im NC: Bearbeitung eines Programms kontinuierlich (Folgesatz) oder satzweise (Einzelsatz).

0 = Folgesatz
1 = Einzelsatz

Datenwort	Name	Richtung	Signalart
011.00-07	Satz ausblenden	HMI → NC	statisch

Mit diesem Signal kann die Funktion ”Satz ausblenden” ein- oder ausgeschaltet werden.

Wirkung im NC: Sätze mit einem vorangestellten ”/” werden ausgeblendet, d. h. sie werden nicht ausgeführt.

0 = Satz ausblenden inaktiv
1 = Satz ausblenden aktiv

Datenwort	Name	Richtung	Signalart
028.00-07	Verfahrtaste Achse 0	HMI → NC	statisch
028.08-15	Verfahrtaste Achse 1	HMI → NC	statisch
029.00-07	Verfahrtaste Achse 2	HMI → NC	statisch
029.08-15	Verfahrtaste Achse 3	HMI → NC	statisch
030.00-07	Verfahrtaste Achse 4	HMI → NC	statisch

Datenwort	Name	Richtung	Signalart
030.08-15	Verfahrtaste Achse 5	HMI → NC	statisch
031.00-07	Verfahrtaste Achse 6	HMI → NC	statisch
031.08-15	Verfahrtaste Achse 7	HMI → NC	statisch
032.00-07	Verfahrtaste Achse 8	HMI → NC	statisch
032.08-15	Verfahrtaste Achse 9	HMI → NC	statisch
033.00-07	Verfahrtaste Achse 10	HMI → NC	statisch
033.08-15	Verfahrtaste Achse 11	HMI → NC	statisch

Diese Signale dienen zum Verfahren der Achsen. Für jede Achse steht ein Byte zur Verfügung. Durch Beschreiben der Bytes wird die jeweilige Aktion ausgeführt.

Wirkung im NC:

-100 bis +100: Verfahren mit % der Maximalgeschwindigkeit. Das Vorzeichen gibt die Richtung an.

+/-101: Tippen um Schrittweite (entsprechend DW44 ... 47) auf der Bahn bei fehlender Vorschubfreigabe. Nur bei einer programmierten Verfahrbewegung verfügbar.

+/-102: Referenzpunktfahrt (Vorzeichen bleibt unbeachtet).

+/-103: Nullpunktsetzen (Vorzeichen bleibt unbeachtet).

+/-104: Schrittfahren.

+/-105: Handrad aktiv - Bewertungsfaktor 1.

+/-106: Handrad aktiv - Bewertungsfaktor 3.

+/-107: Handrad aktiv - Bewertungsfaktor 10.

+/-108: Handrad aktiv - Bewertungsfaktor 30.

+/-109: Handrad aktiv - Bewertungsfaktor 100.

+/-110: Verfahrbefehl für SPS-Achse.

+/-111: Verfahren mit der Geschwindigkeit aus P208 ... 223.

Das Vorzeichen gibt die Richtung an (📖 134).

Datenwort	Name	Richtung	Signalart
036	Allgemeiner Override für Achsen	HMI → NC	statisch
037	Override für Spindeldrehzahlen	HMI → NC	statisch
038	Override für Oszillationsgeschwindigkeit	HMI → NC	statisch
039	Override für SPS-Achsen (Zielpunktfahrt)	HMI → NC	statisch

Bestimmung eines vorzeichenbehafteten Bewertungsfaktors für die momentan gültige Verfahrgeschwindigkeit. Die Angabe erfolgt in 0,1 % Schritten.

Wirkung im NC: Der ausgewählte Wert wird als Faktor zur Bewertung der momentan programmierten Maximalgeschwindigkeit vom Grobinterpolator verwendet.

Der Override ist sowohl im Handbetrieb als auch Automatik wirksam. Er kann durch M-Funktionen innerhalb eines Programms gesperrt werden.

Datenwort	Name	Richtung	Signalart
040-043	Geschwindigkeitsvorwahl für Verfahrtasten	HMI → NC	statisch

Bestimmung der Geschwindigkeit beim modalen Fahren mit Verfahrtasten. Angabe in [mm/min] als 64-Bit floating point Zahl.

Wirkung im NC: Der ausgewählte Wert wird in das Parameterfeld an eine fest definierte Stelle übernommen. Bei jedem Auslösen der Verfahrtaste wird die jeweilige Achse modal mit der eingegebenen Geschwindigkeit gefahren.

Datenwort	Name	Richtung	Signalart
044-047	Schrittweitenvorwahl	HMI → NC	statisch

Bestimmung der Schrittweite beim Handverfahren im Schrittbetrieb. Angabe in [mm] bzw. [inch] als 64-Bit floating point Zahl.

Wirkung im NC: Der ausgewählte Wert wird in das Parameterfeld an eine fest definierte Stelle übernommen. Bei jedem Auslösen der Verfahrtaste wird die jeweilige Achse um das angegebene Delta-Stück bewegt.

6 ETC-MMI-Gateway

Das MMI-Gateway ist das Kommunikationsprogramm zwischen Windows Applikationen und ETC-Steuerungen. Unterschiedliche Applikationen, wie MMIs, Konfigurationstools oder OPC-Server können gleichzeitig ihre Verbindungen zu einer oder mehreren Steuerungen aufbauen.

Das Gateway implementiert alle nötigen Mechanismen für Zugriffsverwaltung, Fehlerhandling und Diagnose und unterstützt steuerungsspezifische Hardwaretreiber und Kommunikationsprotokolle. Mit Hilfe des Konfigurationstools können neue Verbindungen eingefügt bzw. vorhandene Verbindungen editiert werden.

6.1 ETC-MMI-Gateway installieren

Das ETC-MMI-Gateway wird bei der Installation des Lenze-ETC-MMIs mitinstalliert. Es kann aber auch als separate Applikation installiert werden und steht dann als Kommunikationskanal für andere Applikationen zur Verfügung.

Folgende Dateien werden bei der Installation in das Systemverzeichnis auf den PC kopiert (z. B. c:\Windows\System32):

Datei	Beschreibung
mmigtway.exe	Gateway-Programm
mmictr.dll	Schnittstelle für die Applikation
ipcom.dll	DLL mit internen Gateway-Funktionen
gtwconf.exe	Konfigurationsoberfläche
mmigtwayini	Konfigurationsdatei 📖 277

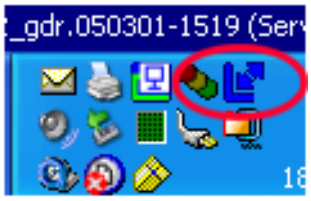
Hinweis!

Wenn Sie das ETC-MMI-Gateway ohne ETC-MMI installieren möchten, kopieren Sie die Dateien manuell in Ihr Systemverzeichnis.

6.2 ETC-MMI-Gateway starten

Das ETC-MMI-Gateway wird automatisch gestartet, wenn eine Applikation (z. B. ETC-MMI) die ”mmictrl.dll” lädt. Dabei wird die aktuelle Konfiguration aus der Datei ”mmigtway.ini” (📖 277) gelesen und überprüft.

In der Taskleiste erscheint eine Symbol für das ETC-MMI-Gateway. Mit einem Mausklick auf das Gateway-Symbol öffnen Sie ein Menü.



ETCN001

ETCN002

Settings: Konfigurationsoberfläche starten.

About: Versions- und Herstellerinformationen anzeigen.

Exit: Gateway schließen (wenn aktive Verbindungen zu einer Applikation bestehen, wird eine Warnung angezeigt).

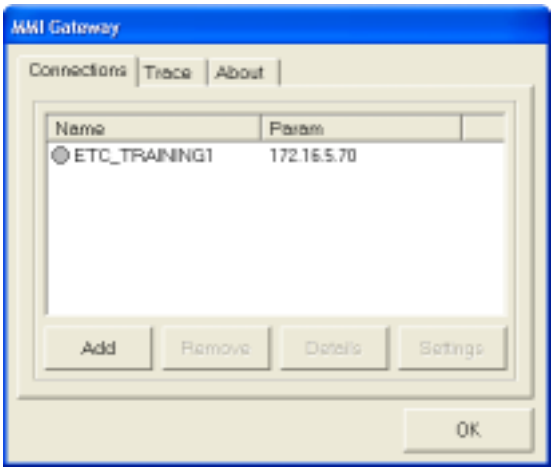
6.3 ETC-MMI-Gateway konfigurieren

Über die Konfigurationsoberfläche des ETC-MMI-Gateways konfigurieren Sie die Verbindungen zu den Steuerungen und rufen Debug-Informationen ab.

Die Konfigurationsoberfläche ist eine separate Applikation (gtwconf.exe), die mit dem ETC-MMI-Gateway installiert wird. Sie kann über das Gateway-Menü (📖 271) oder als Windows-Applikation z. B. über den Windows-Explorer gestartet werden.

6.3.1 Connection - Verbindungen einrichten

Das Register ”Connections” zeigt die Liste der konfigurierten Verbindungen.

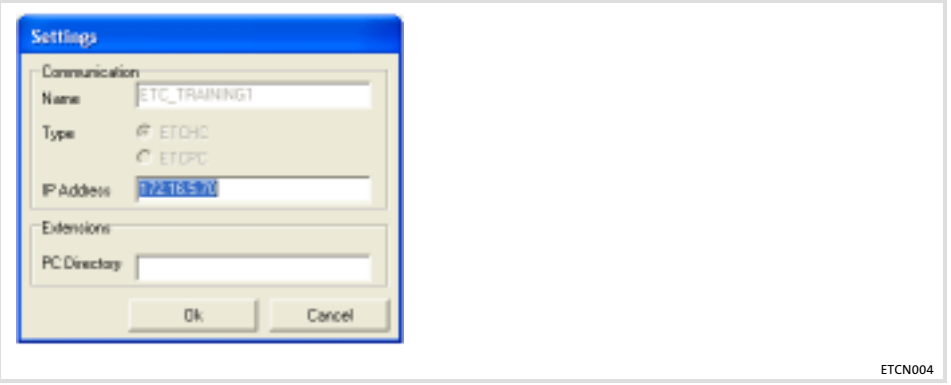


ETCN003

Ein grünes LED-Zeichen neben einem Verbindungsnamen kennzeichnet eine aktive Verbindung, über die Nachrichten und/oder zyklische Daten gesendet werden.

Add - neue Verbindung anlegen

Über die Schaltfläche **Add** legen Sie eine neue Verbindung an. Im Dialog ”Settings” geben Sie die Kommunikationsparameter vor.



Name: Damit eine Applikation über das ETC-MMI-Gateway mit einer Steuerung kommunizieren kann, muss jeder Verbindung ein eindeutiger Name zugewiesen werden. Beliebige Namen sind erlaubt. Vergeben Sie z. B. einheitliche Namen ”ETC0”, ”ETC1” usw. oder applikationsspezifische Namen ”Rampe”, ”Lasersteuerung” usw. Ein Name kann bis 31 ASCII-Zeichen enthalten (”A ... Z”, ”a ... z” , ”0 ... 9”).

Type: Geben Sie hier den Verbindungstyp vor:

- ► UDP-Verbindung für ETCHx-Steuerungen
  Ethernet-Kommunikation über DPR/UDP/IP-Protokoll.
- ► PCI-Verbindung für ETCPx-Steuerungen
  DPR-Kommunikation über einen WDM-Treiber (etcpc.sys, im Lieferumfang der ETCPC).

IP Adress: Geben Sie hier die Verbindungstyp-abhängigen Kommunikationsparameter vor:

- ► ETCHC = IP-Adresse (📖 28)
- ► ETCPC = Index der PCI-Karte (Indizes der PCI-Karten liegen im Bereich 0 ... 9 und werden vom Treiber automatisch zugewiesen. Die erste ETCPC hat den Index 0).

PC Directory:

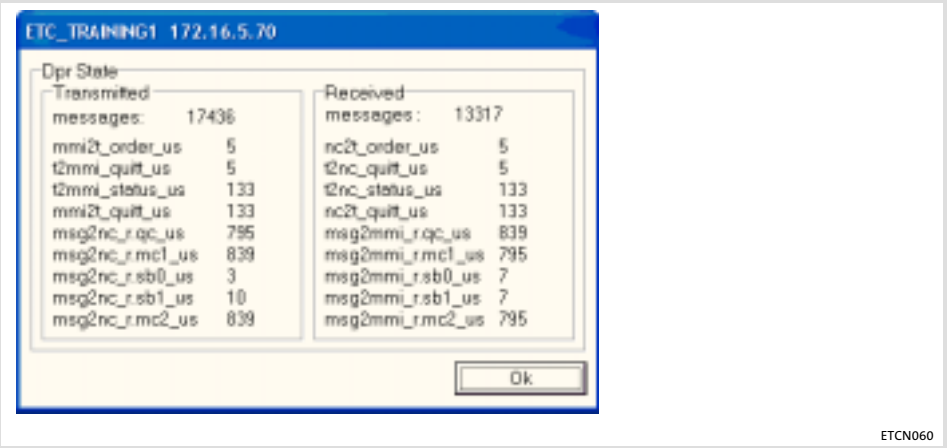
<Ok>: Mit dieser Schaltfläche schreiben Sie die Eingaben in die Datei ”mmigtway.ini” (📖 277). Wenn bereits eine Verbindung mit den gleichen Parametern existiert, wird der Fehler ”Gateway cannot change the parameter” gemeldet.

Remove – Verbindung löschen

Über die Schaltfläche **Remove** löschen Sie die markierte Verbindung. Sie können nur Verbindungen löschen, die von keiner Applikation benutzt werden.

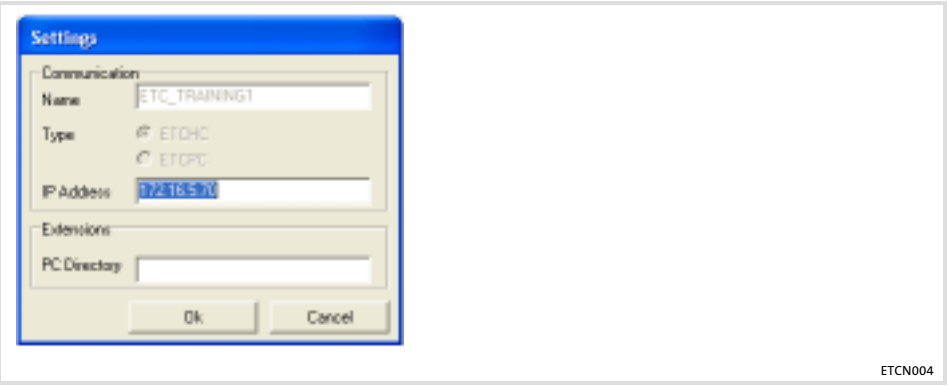
Details – Kommunikationsstatus anzeigen

Wenn Sie sich den Kommunikationsstatus der markierten Verbindung anzeigen lassen möchten, klicken Sie die Schaltfläche **Details**.



Settings – Verbindungsparameter editieren

Über die Schaltfläche **Settings** können Sie die Kommunikationsparameter der markierten Verbindung editiere, wenn sich z. B. die IP-Adresse der Steuerung geändert hat.



Name und **Steuerungstyp** können nicht geändert werden. Möchten Sie auch den Verbindungstyp ändern, löschen Sie die Verbindung und legen eine neue mit demselben Namen an.

IP Adress: Geben Sie hier die Verbindungstyp-abhängigen Kommunikationsparameter vor:

- ETCHC = IP-Adresse (📖 28)
- ETCPC = Index der PCI-Karte (Indizes der PCI-Karten liegen im Bereich 0 ... 9 und werden vom Treiber automatisch zugewiesen. Die erste ETCPC hat den Index 0).

<Ok>: Mit dieser Schaltfläche schreiben Sie die Eingaben in die Datei "mmigtway.ini" (📖 277). Wenn bereits eine Verbindung mit den gleichen Parametern existiert, wird der Fehler "Gateway cannot change the parameter" gemeldet.

Neue Parameter werden vom Gateway sofort übernommen.

6.3.2 Trace – Fehlerlogbuch

Bei Störungen in der Kommunikation können auf diesem Register Trace-Aufnahmen aktiviert werden. Gateway-Traces stellen interne Ereignisse des Kommunikationskanals zwischen der Applikation und dem Gateway dar und enthalten Informationen über den Kommunikationsablauf bzw. über die Ursachen der aufgetretenen Fehler. Trace-Aufnahmen werden in der Datei ”mmigtway.trc” gespeichert, die jedesmal beim Hochlauf des Gateways neu angelegt wird.

Hinweis!

Wir empfohlen, im normalen Betrieb alle Trace-Aufnahmen zu deaktivieren.

Markieren Sie die jeweiligen Kontrollfelder, um Trace-Informationen bei folgenden Ereignissen aufzuzeichnen:

OnChannelCreate: Erzeugen eines neuen Kommunikationskanal im Gateway.

OnConnect: Öffnen/Schließen eines Kommunikationskanals.

OnRead: Empfangen einer Nachrichten über den Kommunikationskanal.

OnWrite: Senden einer Nachrichten über den Kommunikationskanal.

OnLoadFirmware: Übertragen der Firmware.

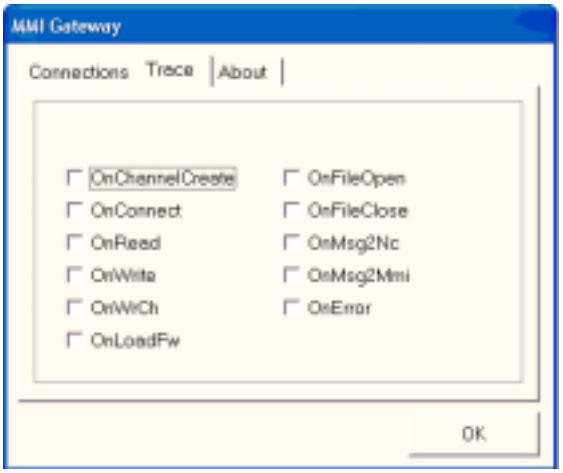
OnFileOpen: Öffnen einer Datei.

OnFileClose: Schließen einer Datei.

OnMsg2Nc: Senden einer Nachricht an die Steuerung.

OnMsg2Mmi: Senden einer Nachricht an die Applikation (ETC-MMI).

OnError: Bei Auftreten eines Fehlers.

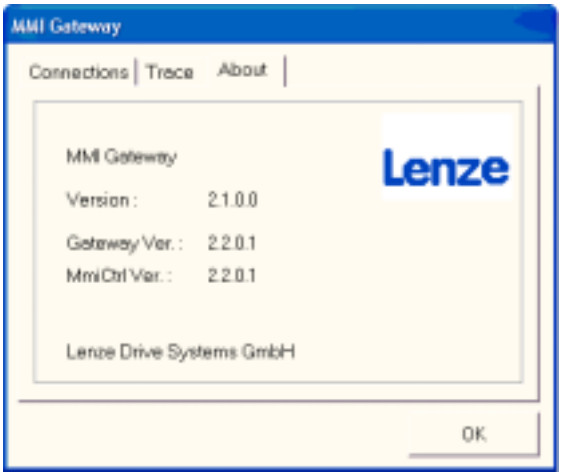


ETCN061

Aktivierte Traces werden beim nächsten Starten des Gateways automatisch wieder eingeschaltet.

6.3.3 About – Versionsinformationen

Das Register ”About” zeigt die Versionsnummern des Gateways, des Konfigurationstools und der MmiCtrl.dll.



ETCN062

6.4 Mmigtway.ini

In der Datei ”mmigtway.ini” wird die aktuelle Konfiguration des Gateways gespeichert.

Sektion [Connection]

In der Sektion [Connections] werden die allgemeine Informationen des Gateways gespeichert. Folgende Schlüsselvariablen sind definiert:

- UdpConns = Anzahl der UDP Verbindungen (Anzahl ETCHCs)
- PciConns = Anzahl der PCI Verbindungen (Anzahl ETCPCs)
- DemoConns = reserviert für simulierte Verbindungen

Sektion [Options]

In der Sektion [Options] mit der Schlüsselvariable ”AutoClose” kann festgelegt werden, ob das Gateway automatisch geschlossen werden soll, wenn keine Verbindung zu einer Applikation besteht.

Sektion [Traces]

Die in der Sektion [Traces] definierten Schlüsselvariablen stellen die aktiven Traces dar. Ist der Wert der Variable ungleich Null werden entsprechende Trace-Informationen in der Datei ”mmigtway.trc” protokolliert (📖 275).

Sektionen der Verbindungen

Für jede Verbindung wird eine weitere Sections angelegt, wobei ”*” der Index der Verbindung ist.:

- [UdpConn*] - für UDP Verbindung (ETCHC)
- [PCIConn*] - für PCI Verbindung (ETCPC)
- [DemoConn*] - für Demo Verbindung

Unter den Schlüsselvariablen ”Name”, ”Param”, ”PcDir” werden die Namen der Verbindung, die Kommunikationsparameter (IP-Adresse für UDP-Verbindungen, Index für PCI-Karten) und der PC-Disk Pfad gespeichert .

6.4.1 Beispiel der Datei ”mmigtway.ini”

```
[Connections]
UdpConns=3
PciConns=1
DemoConns=0
[Options]
AutoClose=0
[Traces]
OnChCreate=0
OnConnect=0
OnRead=0
OnWrite=0
OnWrCh=0
OnLoadFw=0
OnFileOpen=0
OnFileClose=0
OnMsg2Nc=0
OnMsg2Mmi=0
OnError=0
ConnTrc=0
OnAddConn=0
[UdpConn0]
Name=CNC0
Param=172.16.5.113
PcDir=
[UdpConn1]
Name=CNC1
Param=172.16.5.114
PcDir=
[PciConn1]
Name=MyPnc
Param=0
PcDir=
[UdpConn2]
Name=CNC2
Param=172.16.5.115
PcDir=
```

6.5 Kommunikationswerte im DPR Bereich

Name	Bedeutung
Kommunikationsbereich MMI->NC	
mmi2t_order_us	Befehl an Kommunikationsprozessor
t2mmi_quitt_us	Quittung vom Kommunikationsprozessor
t2mmi_status_us	Status vom Kommunikationsprozessor
mmi2t_quitt_us	Quittung an Kommunikationsprozessor
msq2nc_r.qc_uc	Nachrichten-Quittungszähler
msq2nc_r.mc1_us	Nachrichten-Start-Meldungszähler
msq2nc_r.sb0_us	Nachrichten-Hauptgruppe
msq2nc_r.sb1_us	Nachrichten-Untergruppe
msq2nc_r.mc2_us	Nachrichten-Ende-Meldungszähler
Kommunikationsbereich NC-> MMI	
nc2t_order_us	Befehl an Kommunikationsprozessor
t2nc_quitt_us	Quittung vom Kommunikationsprozessor
t2nc_status_us	Status vom Kommunikationsprozessor
nc2t_quitt_us	Quittung an Kommunikationsprozessor
msq2mmi_r.qc_uc	Nachrichten-Quittungszähler
msq2mmi_r.mc1_us	Nachrichten-Start-Meldungszähler
msq2mmi_r.sb0_us	Nachrichten-Hauptgruppe
msq2mmi_r.sb1_us	Nachrichten-Untergruppe
msq2mmi_r.mc2_us	Nachrichten-Ende-Meldungszähler

7 ETC-MMI

Das Programm ”ETC-MMI” wird für folgende Aufgaben verwendet:

- ► Steuerung konfigurieren
- ► Steuerung bedienen und beobachten
- ► Steuerung warten und Fehler diagnostizieren

7.1 ETC-MMI installieren

Stop!

Die PCI-Steuerungsvariante ETCPx erst nach der ETC-MMI-Installation und vor dem ersten Starten des ETC-MMIs installieren.

Hinweis!

Das ETC-MMI-Gateway wird bei der Installation des Lenze ETC-MMIs mitinstalliert

1. Starten Sie über den Windows-Explorer das Programm ”setup.exe” auf der ETC-MMI-Installations-CD.
2. Folgen Sie den Anweisungen des Installationsprogramms. Folgendes wird abgefragt:
   – Demo- oder Standard-Installation
   – Steuerungstyp (ETCHC oder ETCPC)
   – IP-Adresse der ETCHC (📖 28) bzw. Index der ETCPC (Indizes der PCI-Karten liegen im Bereich 0 ... 9 und werden vom Treiber automatisch zugewiesen. Die erste ETCPx hat den Index 0).

   In der Startleiste wird der Eintrag ”Lenze ETC-MMI” angelegt, über den die Applikation gestartet werden kann.
3. Bei einer ETCHC: Überprüfen Sie , dass TCP/IP-Protokolle in Netzwerkeigenschaften installiert sind (**Start → Einstellungen → Systemsteuerung → Netzwerk**).

   Bei einer ETCPC: Bauen Sie die ETCPC-Steckkarte in den PC ein. Beachten Sie dabei die Hinweise im ETC-Gerätehandbuch.

Installierte Dateien

Bei der Standard-Installation befinden sich nach erfolgreicher Installation folgende Dateien und Dateipfade auf der Festplatte des PC:

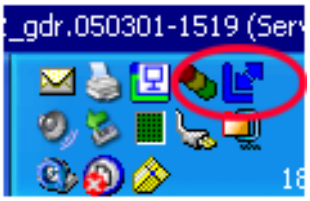
z. B. c:\Programme\Lenze\ETC\MMI\)	
Lenze.exe	ETC-MMI-Applikation
ncform.hlp	Hilfedatei
install.hlp	Installationsanleitung
...\cfg	Konfigurationsverzeichnis
muster.mk	Maschinenkonstanten-Datei mit Grundeinstellungen
mk.hlp	ASCII-Datei, die Hilfsinformationen zur Einstellung der Maschinenkonstanten enthält (”MK ändern”)
mmi_fehl.db	ASCII-Datei, die Fehlermeldungen des MMI enthält
ncr_fehl.db	ASCII-Datei, die Fehlermeldungen des NC enthält
Lenze.txt	ASCII-Datei, mit allgemeinen Texten, Hinweisen, Fehlermeldungen, Informationen über Ein-/Ausgänge und Funktionstasten. Die Datei wird von der Bedienoberfläche benötigt.
sps_fehl.db	ASCII-Datei, die Fehlermeldungen der SPS enthält
delphmmi.ini	Initialisierungs- bzw. Konfigurationsdatei für ETC-MMI
...\bin	
etc*.rsc	ETCHC oder ETCPC Firmware
spsdummy.prg	Beispiel eines SPS-Programms
...\prg	DIN-Datei-Verzeichnis
nikolaus.din	Beispiel einer DIN-Datei (Kontur)
...\log	Ordner für alle temporären Dateien, die von der Steuerung zurückgeladen werden
z. B. c:\Windows\Systems32	
mmigrp32.dll	MMI-DLL
mmigtway.exe	Gateway-Programm
mmictr.dll	Schnittstelle für die Applikation
ipcom.dll	DLL mit internen Gateway-Funktionen
gtwconf.exe	Konfigurationsoberfläche
mmigtwayini	Konfigurationsdatei 📖 277

7.2 ETC-MMI starten

1. Starten Sie das ETC-MMI über **<Start> → Programme → Lenze → ETC**

ETCN011

Das ETC-MMI-Gateway wird automatisch mitgestartet. Die Applikation ist auf der Taskleiste zu sehen:



ETCN001

7.3 ETC-MMI bedienen

7.3.1 Anzeigeelemente der Programmoberfläche

Betriebsarten: Hier steht die aktuell gewählte Betriebsart (”Einrichten”, ”Automatik”, ”Programmieren” oder ”Diagnose”) und, falls angewählt, eine sogenannte Unterbetriebsart.

Hinweis!

Welche Betriebsart nach dem Starten der Bedienoberfläche angezeigt wird, kann vorgegeben werden (📖 320).

Istwertanzeige: Dieser Bereich zeigt die Ist- und Sollpositionen der konfigurierten Achsen.

Statusanzeige: Die Statusfelder dienen zur Anzeige aktueller Programmzustände, wie z. B. die aktuelle Programm-, T-, S-, und M-Nummer. Weiterhin werden hier Informationen zum aktuell bearbeiteten Teileprogramm und SPS-Fehlermeldungen angezeigt.

Softkeys: Am unteren und am rechten Fensterrand befinden sich die horizontalen und vertikalen Softkeys. Deren Beschriftung und Funktion ist abhängig von der angewählten Betriebsart bzw. Unterbetriebsart.

Eingabezeile: Die Eingabezeile dient für spezielle Eingaben. Ist die Zeile aktiv, wird sie weiß dargestellt.

Statuszeile: Hier werden Hinweise und allgemeine Stati angezeigt.

Balkenanzeige: Außer der Standard-Balkenanzeige für "Override" und "V_{ist}" können weitere Anzeigen konfiguriert werden.

7.3.2 Bedienelemente der Programmoberfläche

Die Standard-Bedienoberfläche ist nicht auf eine externe Maschinensteuertafel für den Betrieb der Maschine angewiesen. Alle wichtigen Funktionen (Start, Stopp, Verfahrtasten) sind auf Softkeys (Funktionstasten) gelegt; das heißt, alle Bedienelemente der Programmoberfläche können über die Tastatur aktiviert werden.

Hinweis!

In Anlehnung an die Eigenheiten eines Touch Screens, ist eine Bedienung über Zeigegeräte (z. B. Maus) zwar möglich, die Aktion muss aber mit der **<Enter>**-Taste ausgelöst werden. Das Auslösen von Aktionen über "Doppelklick" ist im Allgemeinen nicht vorgesehen.

Bei entsprechender Programmierung der internen SPS können die Maschinenfunktionen auf separate Tasten gelegt werden, um diese parallel oder alternativ zu den Softkeys auszuwerten.

Stop!

Das Auslösen von Maschinenbewegungen über Softkeys ist nicht an allen Maschinen zulässig ist. Der Betreiber der Maschine muss sicherstellen, dass die gültigen Sicherheitsrichtlinien eingehalten werden.

Betriebsarten-unabhängige Bedienelemente

Am oberen Fensterrand wechseln Sie über die Softkeys **Einrichten**, **Automatik**, **Programm** oder **Diagnose** bzw. über die Tasten **<F9>** ... **<F12>** die Betriebsart des ETC-MMIs. Über den Softkey **SPS-Tasten** oder mit **<Shift>+<F8>** schalten Sie die Softkeyleiste auf "SPS-Tasten" um. Diese müssen in der SPS programmiert sein (📖 73).

Betriebsarten-abhängige Bedienelemente

Eine Beschreibung der Betriebsarten-abhängigen Softkeys entnehmen Sie den jeweiligen Beschreibungen der Betriebsarten weiter hinten in diesem Kapitel. Im Folgenden gehen wir auf das unterschiedliche Verhalten von Softkeys beim Betätigen ein:

Direktes Auslösen einer Funktion (Taster)

Die zugeordnete Funktion wird ausgeführt, beim Drücken bzw. Loslassen der Taste oder solange die Taste gedrückt gehalten wird.

ETCN065

Aktivierung eines bestimmten Zustands (Schalter)

Die zugeordnete Funktion wird nach Drücken der Taste solange ausgeführt, bis dieselbe Taste erneut gedrückt wird. Solange die Funktion aktiv ist, wird die Taste ”gedrückt” dargestellt. Der aktive Zustand wird im Folgenden als Unterbetriebsart bezeichnet und in der Betriebsartenzeile angezeigt.

ETCN067

Einblenden einer neuen Softkeyzeile (Untermenü)

Diese Funktionstasten öffnen ein sogenanntes Untermenü; das heißt, sie verändern die Beschriftung und damit die Bedeutung anderer Tasten. Die Beschriftung dieses Funktionstastentyps endet immer mit drei Punkten.

ETCN066

Über die Funktionstaste **<Zurück>** wechseln Sie wieder zurück in die aufrufende Tastenebene.

7.3.3 Hilfefunktion

Mit der Tastenkombination **<STRG>+<SHIFT>+<beliebige F-Taste>** starten Sie die Onlinehilfe.

7.3.4 Konfigurationsdatei

Die Konfigurationsdatei ”delphmmi.ini” enthält Einstellungen, die zum Betrieb des MMIs notwendig sind und darf nur von geschultem Personal geändert werden. Weitere Informationen (📖 320).

7.3.5 Sprachumschaltung

Alle Texte der Bedienoberfläche werden in einer ASCII-Datei gespeichert, die mit einem beliebigen Editor geändert werden kann. Auf diese Weise können Texte individuell angepasst oder in Fremdsprachen übersetzt werden (im Rahmen der darstellbaren Schriftzeichen). Die Sprachdatei wird in der Konfigurationsdatei ”delphmmi.ini” eingestellt. Weitere Informationen (📖 314).

7.3.6 Passworte

Im ETC-MMI kann für jede Betriebsart und zusätzlich für das Ändern der Passworte ein eigenes Passwort vergeben werden. Diese werden nach dem Start des ETC-MMIs oder bei einem Betriebsartenwechsel abgefragt. Voreingestellt sind folgende Passworte:

Betriebsart / Passwortverwaltung	Passwort
Einrichten	1
Automatik	2
Programmieren	3
Diagnose	4
Passwort ändern	5

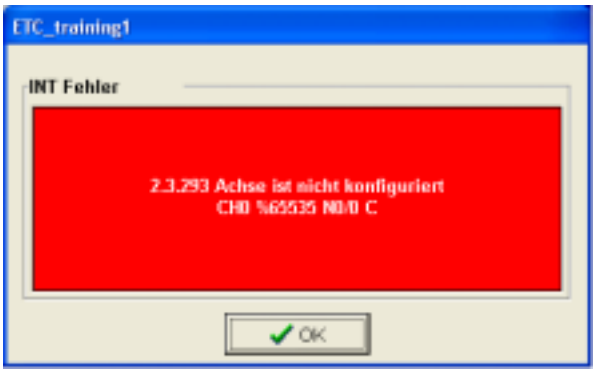
Passwörter ändern Sie in der Betriebsart ”Diagnose” (📖 306). Sie werden verschlüsselt in der Datei ”Lenze.pwd” gespeichert.

Hinweis!

Wenn Sie im Ordner ”cfg” die Datei ”Lenze.pwd” löschen, werden alle Passworte wieder auf die in der Tabelle genannten Standards zurückgesetzt. Wenn Sie keine Passwortabfrage wünschen (z. B. während der Inbetriebnahme), geben Sie als Passwort ein ”Leerzeichen” ein.

7.3.7 Hinweise, Warnungen, Fehlermeldungen

Hinweise, Warnungen und Fehler, die das NC-System meldet, werden in einem separaten Fenster angezeigt. In der Titelzeile steht die Bezeichnung der Steuerung, die den Fehler meldet.



ETCN064

Ein solches Fenster muss mit **<Enter>** quittiert werden, bevor andere Tastatureingaben angenommen werden (modaler Dialog). Die Mitteilungstypen werden anhand ihrer Farbe unterschieden. Dabei stehen ein blauer Hintergrund für einen Hinweis, ein gelber Hintergrund für eine Warnung und ein roter Hintergrund für einen Fehler. Alle Mitteilungen werden in eine Log-Buch-Datei mit Datum und Uhrzeit des Vorkommnisses auf dem PC gespeichert und sind in der Betriebsart ”Diagnose” zugänglich (📖 306).

7.4 Betriebsart ”Einrichten”

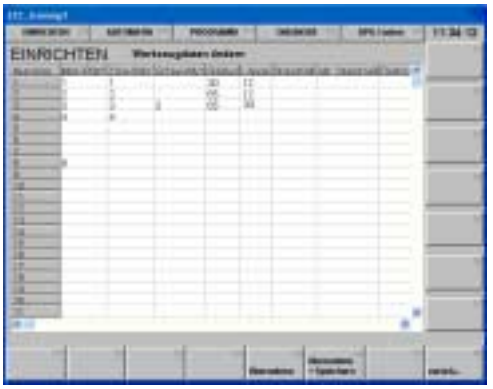
ETCN068

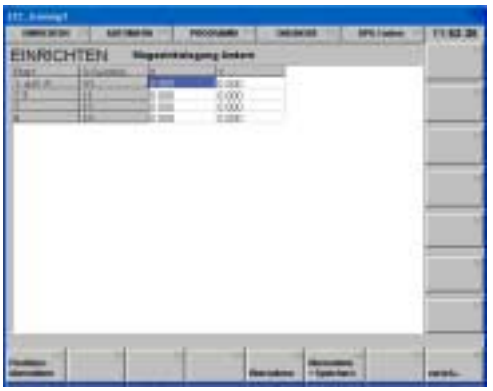
Die Betriebsart ”Einrichten” beinhaltet Funktionen zum Einrichten der Anlage. Unter anderem können Sie hier Referenz- und Handfahrten durchführen und Werkzeuge verwalten.

Horizontale Funktionstasten

Referenz...	
automatisch	Über die **<Start>**-Taste kann eine automatische Referenzpunktfahrt eingeleitet werden. Alle Achsen werden in der konfigurierten Reihenfolge referenziert.
manuell	Eine manuelle Referenzpunktfahrt kann eingeleitet werden. Sie wählen die gewünschte Achse mit der Taste **<Achse>** (oder <Cursor>-Tasten) und verfahren die Achsen mit der Taste **<Fahren +>**/**<Fahren ->** oder **<Start>**.
zurück...	Zurück zur vorherigen Ebene.
Handfahren...	Hier können Sie die Achsen manuell verfahren.
modales Fahren	Die mit der Taste **<Achse>** (oder <Cursor>-Tasten) angewählete Achse wird verfahren, solange die Taste **<Fahren +>**/**<Fahren ->** betätigt wird. Die Geschwindigkeit, mit der die Achsen verfahren werden, ist durch MK_MODVMAX vorgegeben, kann aber mit der Taste **<Override +>**/**<Override ->** beeinflusst werden.
Schrittfahren	Die angewählte Achse verfährt um die eingegebene Schrittweite (s. u.). Die Richtung wird durch die Tasten **<Fahren +>**/**<Fahren ->** bestimmt. Beim ”Schrittfahren” verfahren die Achsen im Eilgang.
Schrittweite	Schrittweite in mm, um die die angewählte Achse verfahren soll. Die Eingabe muss mit **<ENTER>** abgeschlossen werden. Die Funktionstaste wird erst beschriftet und freigegeben, wenn vorher die Taste **<Schrittfahren>** betätigt wurde.

Zielpunktfahren	Die Funktion ”Zielpunktfahren” bezieht sich auf alle Achsen. Wenn Sie die Taste **<Start>** drücken, werden alle Achsen verfahren, deren Ist- und Soll-Position nicht identisch sind. Beim ”Zielpunktfahren” verfahren die Achsen im Eilgang.
Zielposition	Zielposition für die ausgewählte Achse. Sie wählen die gewünschte Achse mit der Taste **<Achse>** (oder <Cursor>-Tasten). Die Eingabe muss mit **<ENTER>** abgeschlossen werden. **<ESC>** bricht die Eingabe ab und stellt die alten Positionen wieder ein. Die Funktionstaste wird erst beschriftet und freigegeben, wenn vorher die Taste **<Zielpunktfahren>** betätigt wurde.
Handrad	**Die „Handrad-Tasten“ werden nur angezeigt, wenn in den Maschinenkonstanten ein Handrad konfiguriert ist.** Verfahren der Achsen über ein Handrad. Sie wählen die Achse, die Sie mit dem Handrad verfahren möchten, mit der Taste **<Achse>** (oder <Cursor>-Tasten).
Handrad-Faktor	Auswahl eines Faktors, mit dem die Verfahrgeschwindigkeit beaufschlagt wird. 5 Handradfaktoren sind vorgesehen (1, 3, 10, 30, 100). Bei jeder Betätigung der Taste **<Handrad-Faktor>** wird auf den nachfolgenden Faktor umgeschaltet. Die Funktionstaste wird erst beschriftet und freigegeben, wenn vorher die Taste **<Handrad>** betätigt wurde.
zurück...	Zurück zur vorherigen Ebene.
Nullpunkte...	
Nullen	Die Ist-Position der angewählten Achse wird auf Null gesetzt (Nullpunktverschiebung).
Setzen	Die Ist-Position der angewählten Achse wird auf den eingegebenen Wert gesetzt.
Korrigieren	Die Ist-Position der angewählten Achse wird um den eingegebenen Wert verschoben.
S umschalten	Eingabe der gewünschten Nummer des Werkstück-Koordinatensystems (S), Abschluss mit **<Enter>**, Abbruch mit **<ESC>**. Das eingegebene Koordinatensystem wird aktiviert.
zurück...	Zurück zur vorherigen Ebene.
Werkzeugverwaltung...	Verwaltung von Werkzeugen und den zugehörigen Wechselpositionen. Bis zu 99 Werkzeugen können eingegeben und gespeichert werden. Werkzeug ”0” dient als Referenzwerkzeug und sollte nicht anderweitig verwendet werden. Den Werkzeugen können 99 Werkzeugmagazinplätze zugeordnet werden. Beachten Sie, dass jeder Magazinplatz die Anzahl der Werkstückkoordinatensysteme (max. 99) verringert. Die Anzahl der Werkzeuge und Magazinplätze, die im MMI verwaltet werden, sind in der ”DelphMMI.ini” festgelegt (📖 320). Für jedes Werkzeug stehen 30 Parameter zur Verfügung. Davon haben sechs Parameter eine feste Bedeutung: ● Werkzeuglänge ● Werkzeugradius ● Standzeit ● Drehzahl ● Magazinplatz ● Schwesterwerkzeug Die Daten werden bei Anwahl des Werkzeugs im Teileprogramm im P-Feld zur Verfügung gestellt (P700 ... P729).

Werkzeugdaten...	Mit dieser Taste werden die aktuellen Werkzeugkorrekturdaten vom NC-Rechner angefordert und in der Datei ”vom_nc.wtk” im Konfigurationsverzeichnis gespeichert. Die empfangenen Daten werden gemäß der Spezifikation der Werkzeugverwaltung interpretiert und in der Werkzeugkorrekturtabelle angezeigt. Jedem definierten Werkzeug kann ein Magazinplatz zugeordnet werden. Eine Mehrfachvergabe eines Magazinplatzes ist nicht zulässig. Ein Werkzeug gilt dann als definiert, wenn im Feld ”Originalwerkzeug” eine Werkzeugnummer eingetragen ist. Ein Schwesterwerkzeug legen Sie an, indem Sie beim Schwesterwerkzeug im Feld ”Originalwerkzeug” die Nummer des Originalwerkzeugs eintragen. Dabei besteht die Möglichkeit, Daten, die bereits beim Originalwerkzeug eingetragen wurden, in das Schwesterwerkzeug zu übernehmen. Vor Verlassen eines Feldes erfolgt eine Überprüfung der eingegebenen Werte. Nur wenn korrekte Werte eingegeben wurden, kann das Feld auch tatsächlich verlassen werden. Beim Ändern der Felder ”Originalwerkzeug” und ”Schwesterwerkzeug” werden Felder teilweise automatisch ausgefüllt.
	 ETCN069
Übernahme	Die Daten aus der Tabelle werden in den NC-Rechner übernommen; allerdings nur temporär, d. h. die Informationen gehen beim Ausschalten der Steuerung verloren.
Übernahme + Speichern	Die Daten aus der Tabelle werden in den NC-Rechner übertragen und auf dem PC in einer Datei gespeichert.
zurück...	Zurück zur vorherigen Ebene.
Wechselpositionen...	Mit dieser Taste werden die aktuellen Magazindaten vom NC-Rechner angefordert und in der Datei ”vom_nc.wsk” im Konfigurationsverzeichnis gespeichert. Die empfangenen Daten werden gemäß der Spezifikation der Werkzeugverwaltung interpretiert und angezeigt. Die Tabelle stellt die Zuordnung zwischen Magazinplätzen und Koordinatensystemen dar. Die Wechselpositionen werden ebenfalls in der Tabelle eingetragen. Die Positionen für X, Y, Z und C können entweder von Hand eingetragen oder über eine Teachfunktion ermittelt werden. Wenn bei der Maschine die Möglichkeit besteht, die Achsen unabhängig von Softkeys zu bewegen, können die Positionen angefahren werden und mit dem Softkey **<Positionsübernahme>** für den angewählten Magazinplatz übernommen werden. Die Daten werden in die entsprechenden Spalten eingetragen.
Positionsübernahme	Die aktuellen Achspositionen werden als Werkzeugwechselpositionen eingelesen und in die Tabelle übernommen.

Übernahme	Die Daten aus der Tabelle werden in den NC-Rechner übernommen; allerdings nur temporär, d. h. die Informationen gehen beim Ausschalten der Steuerung verloren.  ETCN070
Übernahme + Speichern	Die Daten aus der Tabelle werden in den NC-Rechner übertragen und auf dem PC in einer Datei gespeichert.
zurück...	Zurück zur vorherigen Ebene.
Wkz-Satz laden	Dialog öffnen, in dem eine Datei mit Werkzeugdaten (WTK Datei) ausgewählt werden kann. Diese wird dann zur Steuerung übertragen.
Werkzeugnummer	Im Dialog ”Statusanzeigen” befindet sich am oberen Rand ein Feld, in dem eine Werkzeugnummer (Tx) eingegeben werden kann. Nach Betätigung der Taste erscheint das Feld farblich hinterlegt und der Cursor blinkt in diesem Feld. Hier kann die Werkzeugnummer angegeben werden. Änderungen werden mit **<Enter>** übernommen und mit **<ESC>** verworfen.
zurück...	Zurück zur vorherigen Ebene.
M-Funktion eingeben	Im Dialog ”Statusanzeigen” befindet sich am oberen Rand ein Feld, in dem eine M-Funktion eingegeben werden kann. Nach Betätigung der Taste erscheint das Feld farblich hinterlegt und der Cursor blinkt in diesem Feld. Hier kann die Funktionsnummer angegeben werden. Änderungen werden mit **<Enter>** übernommen und mit **<ESC>** verworfen. Die M-Funktion wird an die SPS zur Ausführung übergeben.
Eingabe	Einzelsatz (z. B. G-, S-, T- und M-Funktionen) eingeben. Die Ausführung wird mit **<Enter>** abgeschlossen und damit ausgelöst.

Vertikale Funktionstasten

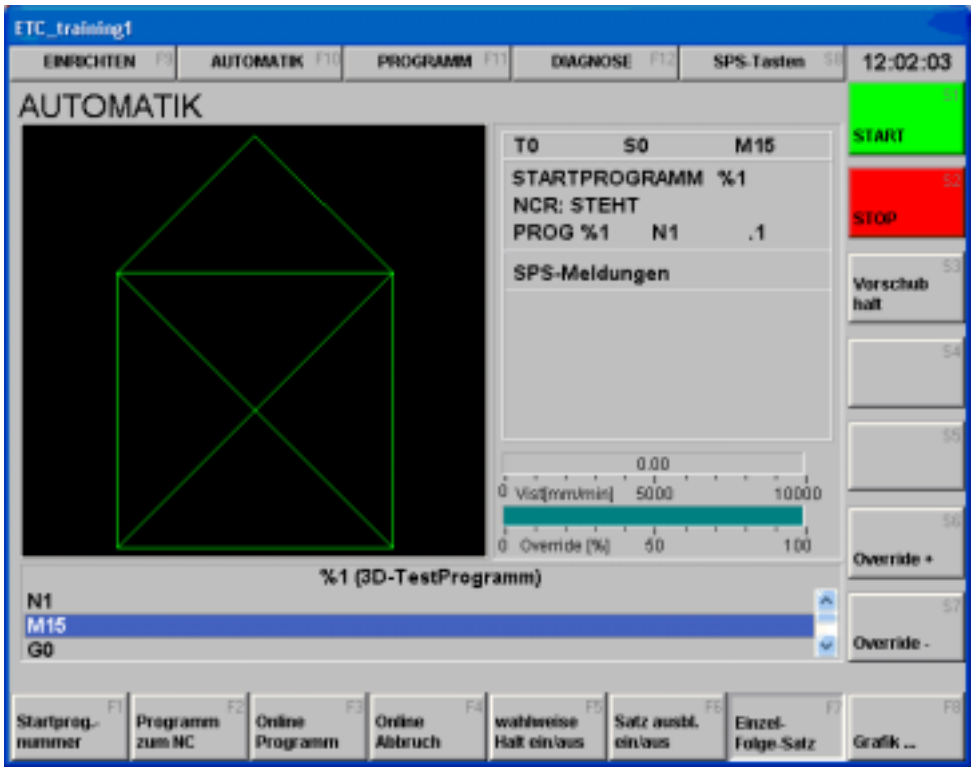
Start	Aktion (z. B. automatische Referenzpunktfahrt) auslösen.
Stop	Alle Achsbewegungen sowie ein laufendes Programm beenden.
Fahren+	Handverfahrtaste für positive Achsrichtung der ausgewählten Achse.
Fahren-	Handverfahrtaste für negative Achsrichtung der ausgewählten Achse.
Achse	Achse, die verfahren werden soll, auswählen. Alternativ kann die Achse mit den **<Cursor>**-Tasten angewählt werden.
Override+	Achs- bzw. Bahngeschwindigkeit der angewählten Achse vergrößern (in Prozent von MK_VMAX). Der Wert wird oberhalb von 10 % in 10 %-Schritten verändert und unterhalb von 10 % in 1 %-Schritten.
Override-	Achs- bzw. Bahngeschwindigkeit der angewählten Achse verkleinern (in Prozent von MK_VMAX). Der Wert wird oberhalb von 10 % in 10 %-Schritten verändert und unterhalb von 10 % in 1 %-Schritten.

Vertikale Funktionstasten im Menü Werkzeugverwaltung

Beim Umschalten in die Werkzeugverwaltung wird eine neue vertikale Softkeyzeile eingeblendet.

Werkzeug holen	Das in den Statusfeldern angezeigte Werkzeug ”Tx” wird aus dem Werkzeugmagazin geholt und in den Werkzeughalter eingesetzt.
Werkzeug wechseln	Das in den Statusfeldern angezeigte Werkzeug ”Tx” wird gegen das momentan im Werkzeughalter befindliche Werkzeug getauscht. Die Funktionen des Werkzeugwechsels erfordern ein entsprechendes SPS-Prgramm (DIN-Programme).
Werkzeug ablegen	Das in den Statusfeldern angezeigte Werkzeug ”Tx” wird im Werkzeugmagazin abgelegt.

7.5 Betriebsart ”Automatik”



ETCN071

Die Betriebsart ”Automatik” ist immer dann eingestellt, wenn sich der NC-Rechner in einer regulären Programmabarbeitung befindet. Es zeigt die wichtigsten Informationen des Programmablaufs.

Horizontale Funktionstasten

Startprogrammnummer	Programmnummer des zu bearbeiteten Programms eingeben. Das Programm muss zuvor z. B. mit der Funktion ”Programm zum NC” übertragen werden (ggf. mit allen benötigten Unterprogrammen). Das Programm wird durch die **<Start>**-Taste gestartet.
Programm zur NC	Programm vom PC an die ETCxC übertragen. Nach Aufruf der Funktion erscheint ein Dialog. Wählen Sie über die **<Cursor>**-Tasten ein Programm aus und starten Sie die Übertragung mit **<Enter>**. **<ESC>** bricht die Aktion ab. Hinweis: In der ”DelphMMi.ini” können Sie vorgeben, welche Programme beim Initialisieren der Steuerung automatisch geladen werden (📖 320).
Online Programm	Diese Funktion setzen Sie ein, wenn ein Programm so groß ist, dass es nicht auf einmal an die NC gesendet werden kann. Ein Dialog wird geöffnet (📖 300) in dem Sie das gewünschte Programm auswählen. Nach**<Enter>** wird das Programm blockweise an die ETCxC gesendet. Mit **<Start>** wird das Programm gestartet, ohne dass es vollständig übertragen sein muss. Die Ausgabezeile zeigt das ausgewählte Programm und den prozentual übertragenen Anteil. Wenn hinter der Prozentzahl der Text ”WAIT” ausgegeben wird, kann der Speicher in der ETCxC momentan nichts mehr aufnehmen. Wurde der übertragene Block abgearbeitet, wird automatisch der nächste Block gesendet, bis das Programm vollständig abgearbeitet ist. Um ein Programm ”online” zu bearbeiten, ist zu beachten, dass das Programm keine Sprünge und Verzweigungen enthalten darf. Bei Unterprogrammen oder Zyklusaufrufen ist darauf zu achten, dass die Programme vorher in die ETCxC geladen wurden.
Online Abbruch	Online-Modus beenden. Bei einem Neustart muss die Bearbeitung von vorn begonnen werden.
wahlweise Halt ein/aus	Programm bei einem ”M1”-Befehl anhalten. Mit **<Start>** wird das Programm fortgesetzt. M1 muss von der SPS entsprechend bearbeitet werden.
Satz ausblenden ein/aus	Alle mit einem ”/” gekennzeichneten Sätze nicht ausführen.
Einzel-Folge-Satz	Jeder Verfahrsatz muss durch die **<Start>**-Taste angestoßen werden. Bei einer Umschaltung im laufenden Betrieb wird die Bearbeitung mit dem nächst möglichen Satz angehalten. Nach Aufhebung der Funktion ”Einzelsatz” muss die **<Start>**-Taste erneut betätigt werden.
Grafik...	
Grafik ein/aus	Eine graphische Anzeige des aktuellen Programms anstelle der Positionsanzeige einblenden. Die grafische Darstellung beschränkt sich derzeit auf die reine Konturbeschreibung der Achsen X und Y. Nullpunktverschiebungen, Drehungen, individuelle Programmierung und Ähnliches werden nicht unterstützt.

Vertikale Funktionstasten

Start	Startet das ausgewählte Programm bzw. die satzweise Bearbeitung im Einzelsatzbetrieb.
Stop	Beendet unmittelbar die Programmausführung. Alle Achsen werden mit den eingestellten Bremsrampen angehalten. Bei einem erneuten Start wird die Bearbeitung von vorn gestartet.
Vorschub halt	Wird ”Vorschub halt” aktiviert, werden die Achsen nicht weiter verfahren, bis die Funktion wieder inaktiviert wird. Die Achsen werden mit den eingestellten Bremsrampen angehalten. Die aktuelle Verfahrgeschwindigkeit sowie der eingestellte Override-Wert werden im Statusfeld angezeigt. Die Ausführung der Funktion muss über die SPS erfolgen. Entweder ist ”Vorschub halt” aktiv oder ”wahlweise Halt”. ETCN072
Override+	Achs- bzw. Bahngeschwindigkeit der angewählten Achse vergrößern (in Prozent von MK_VMAX). Der Wert wird oberhalb von 10 % in 10 %-Schritten verändert und unterhalb von 10 % in 1 %-Schritten.
Override-	Achs- bzw. Bahngeschwindigkeit der angewählten Achse verkleinern (in Prozent von MK_VMAX). Der Wert wird oberhalb von 10 % in 10 %-Schritten verändert und unterhalb von 10 % in 1 %-Schritten.

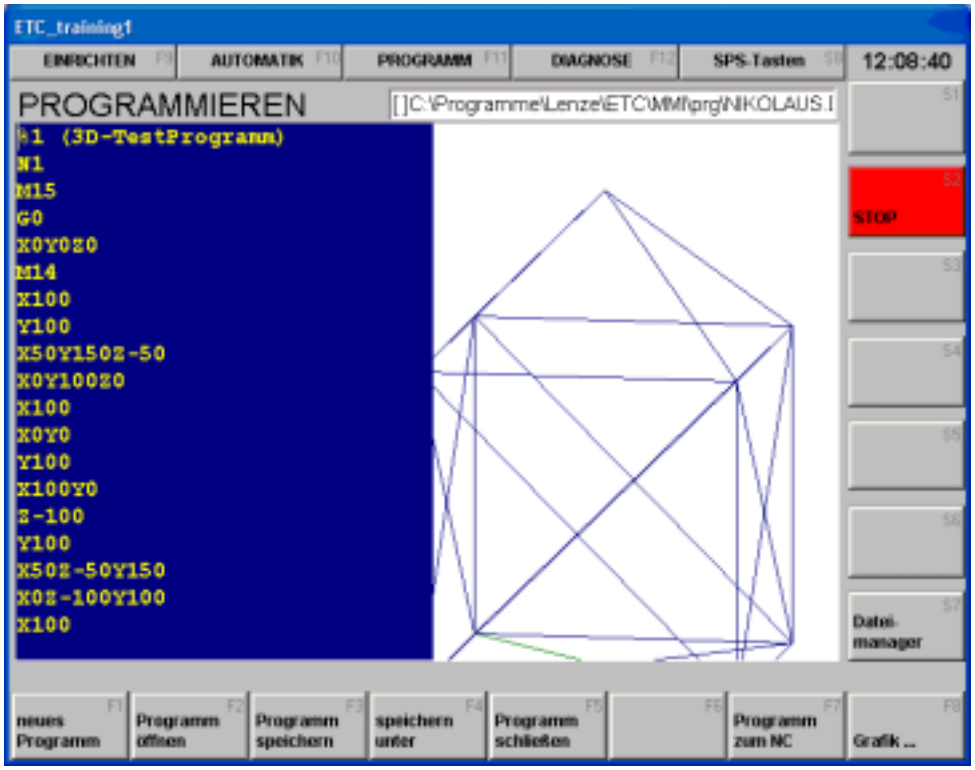
Horizontale Funktionstasten bei Grafik ”ein”

Zeichnung umranden	Ein Rahmen wird um das DIN-Programm gezeichnet und in Eingabeeinheiten bemaßt.
Vergrößern	In die Kontur zoomen.
Verkleinern	Aus der Kontur zoomen.
Zoom	Vollbild-Darstellung der Grafik. Alle anderen Anzeigen sind ausgeblendet.

Vertikale Funktionstasten bei Grafik ”ein”

Leerfahrt anzeigen	Leerfahrten werden andersfarbig dargestellt (M15/M14). Anmerkung: Die Grafik erkennt Leerfahrten an einer vorgeschalteten M15, Arbeitswege an M14.
Fortschritt farbig	Vollständig gefahrene Strecken erhalten einen Farbumschlag.
X plus	Darstellung in X plus Richtung verschieben.
X minus	Darstellung in X minus Richtung verschieben.
Y plus	Darstellung in Y plus Richtung verschieben.
Y minus	Darstellung in Y minus Richtung verschieben.

7.6 Betriebsart ”Programmieren”



ETCN073

In der Betriebsart ”Programmieren” editieren Sie CNC-Programme oder beliebige Textdateien.

Die Betriebsart bietet eine Dateiverwaltung, mit der Dateien kopiert, gedruckt und gelöscht werden können (📖 302) sowie einen ASCII-Editor (📖 299), über den Programme editiert werden können.

Der Name, der in Bearbeitung befindlichen Datei wird oben rechts angezeigt. Im Editorfeld (in Abbildung links) wird der Inhalt der Datei angezeigt und editiert. Bei einem CNC-Programm kann optional eine grafische Darstellung eingeblendet werden (in Abbildung recht).

Bis zu acht Dateien können gleichzeitig geöffnet werden. Die Namen werden in eine Liste eingetragen. Mit der **<TAB>**-Taste und den **<Cursor>**-Tasten werden die Dateien angewählt und mit der **<Enter>**-Taste in den Vordergrund geholt.

Die Bedienung des Editors wird in einem separaten Abschnitt beschrieben .

Horizontale Funktionstasten

neues Programm	Der Editor für die Eingabe einer neuen Datei vorbereiten. Wenn bereits eine andere Datei in Bearbeitung ist, bleibt diese im Hintergrund weiter aktiv. Über die Dateiauswahlzeile im oberen Fensterbereich können Dateien aus dem Hintergrund in den Vordergrund gehlt werden.
Programm öffnen	Bestehende (Programm-)Datei öffnen. Ein Dialog wird geöffnet (📖 300) in dem Sie die gewünschte Datei wählen. Nach**<Enter>** wird die Datei im Editor geöffnet.

Programm speichern	Angezeigte (Programm-)Datei speichern. Wenn die Datei neu angelegt wurde, wird ein Dialog geöffnet (📖 300), in dem Sie Verzeichnis und Dateiname vorgeben.
speichern unter	Angezeigte (Programm-)Datei unter einem anderen Namen speichern. Ein Dialog wird geöffnet (📖 300) in dem Sie Verzeichnis und Dateiname vorgeben. Der Inhalt der Originaldatei bleibt unverändert.
Teach-IN...	Diese Taste ist nur sichtbar, wenn in der SPS ein entsprechendes Bit gesetzt ist. Teach-Modus einschalten. Im Teach-Modus kann die aktuelle Position aufgenommen und an die Cursor-Position im Editor übertragen werden.
Teach-In EIN/AUS	In diesem Modus können die Achsen über die SPS verfahren werden. Mit einer SPS-Nachricht wird die aktuelle Position im Editor als Verfahrsatz mit den Achsbuchstaben der für das Teachen konfigurierten Achsen eingetragen.
Teach-In Erweitert	Im erweiterten Teach-In-Mode ist es möglich, den Programmablauf im Einzelsatzbetrieb im Editor zu verfolgen. Die aktuelle Zeilenposition der ETCxC wird hervorgehoben. Voraussetzung dafür ist, dass das aktuell auf der ETCxC laufende Programm auch im Editor geladen ist. Der Bediener hat nun die Möglichkeit, über den Einzelsatzbetrieb sein Programm zu kontrollieren und ggf. einzelne Sätze ”nachzuteachen“. Auch in diesem Modus gilt, dass die Sätze überschrieben werden, wenn der Cursor auf keiner Leerzeile steht. Andernfalls werden die Sätze eingefügt. **Achtung:** Wenn neue Sätze eingefügt werden, stimmt die Zeilennummer im Editor nicht mehr mit der Zeilennummer im laufenden Programm der ETCxC überein. Das bedeutet, dass im weiteren Ablauf des Programms die falsche Zeilenposition angezeigt wird.
Programm schließen	Programmbearbeitung beenden und Editorinhalt löschen. Die Originaldatei wird nicht verändert.
Programm zur NC	Programm vom PC an die ETCxC übertragen. Nach Aufruf der Funktion erscheint ein Dialog. Wählen Sie über die **<Cursor>**-Tasten ein Programm aus und starten Sie die Übertragung mit **<Enter>**. **<ESC>** bricht die Aktion ab. Hinweis: In der ”DelphMMi.ini” können Sie vorgeben, welche Programme beim Initialisieren der Steuerung automatisch geladen werden (📖 320).
Grafik...	Anzeige (im 3 D-Modus) der Kontur eines Programms, welches sich im Editor befindet. Die grafische Darstellung beschränkt sich derzeit auf die reine Konturbeschreibung der Achsen X, Y und Z. Nullpunktverschiebungen, Drehungen, individuelle Programmierung und ähnliches werden nicht unterstützt.
Reset Bildlage	Ursprüngliche Lage der Kontur anzeigen.
Position anzeigen	Läuft ein Programm, kann durch ”Position anzeigen” die aktuelle Position auf der Kontur dargestellt und verfolgt werden.
Grafik ein/aus	Grafische Anzeige ein- und ausblenden. Die Größe des Fensters kann mit Hilfe der Funktion ”Bildschirmteilung” verändert werden.
zurück...	Zurück zur vorherigen Ebene.

Vertikale Funktionstasten

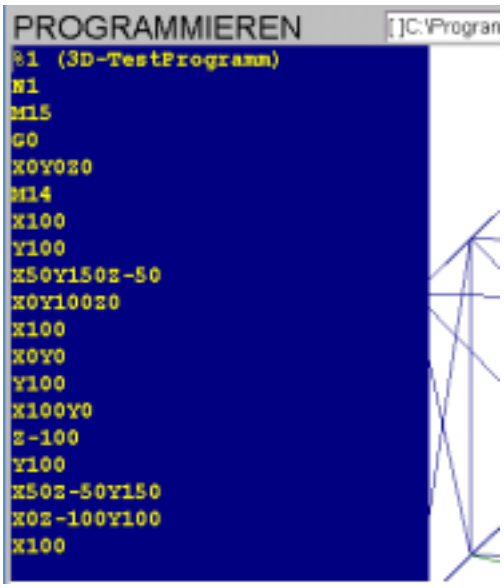
Stop	Beendet ein laufendes Programm.
Zyklus einfügen	Einfügen eines vorbereiteten Zyklus. Mit den **<Cursor>**-Tasten wird der Zyklus ausgewählt und mit **<Enter>** in der Zeile eingefügt, in der der Cursor steht. **<ESC>** bricht die Auswahl ab. Ein Zyklus ist ein Unterprogramm, dem beim Aufruf Parameter übergeben werden. Nähere Informationen zur Zyklenerstellung finden Sie weiter hinten in diesem Kapitel (📖 303.).
Zyklus bearbeiten	Bestehenden Zyklus editieren. Der Cursor muss auf der Zeile stehen, in der der Zyklus aufgerufen wird.
Zyklus löschen	Zyklusaufruf an der aktuellen Cursorposition löschen.
Dateimanager	Die genaue Beschreibung des Dateimanagers inklusive der Bedeutung der dort verwendeten Softkeys finden Sie weiter hinten in diesem Kapitel(📖 302.).

Vertikale Funktionstasten bei Grafik ein

Rotiere um X, Y, Z	Darstellung der Kontur in X-, Y- und Z-Richtung drehen (über Tasten **<+>** und **<->**).
Verschiebe um X, Y	Darstellung der Kontur in X- und Y-Richtungverschieben (über Tasten **<+>** und **<->**).
Bildschirmteilung	Größe des Grafikfensters ändern (über Tasten **<+>** und **<->**).
Zoom	Anzeigegröße der Grafik ändern (über Tasten **<+>** und **<->**).

7.6.1 ASCII-Editor

Der Editor dient in erster Linie zur Eingabe und zum Ändern von CNC-Programmen nach DIN 66025, aber auch zum Editieren beliebiger ASCII-Dateien.



ETC075

Hinweis!

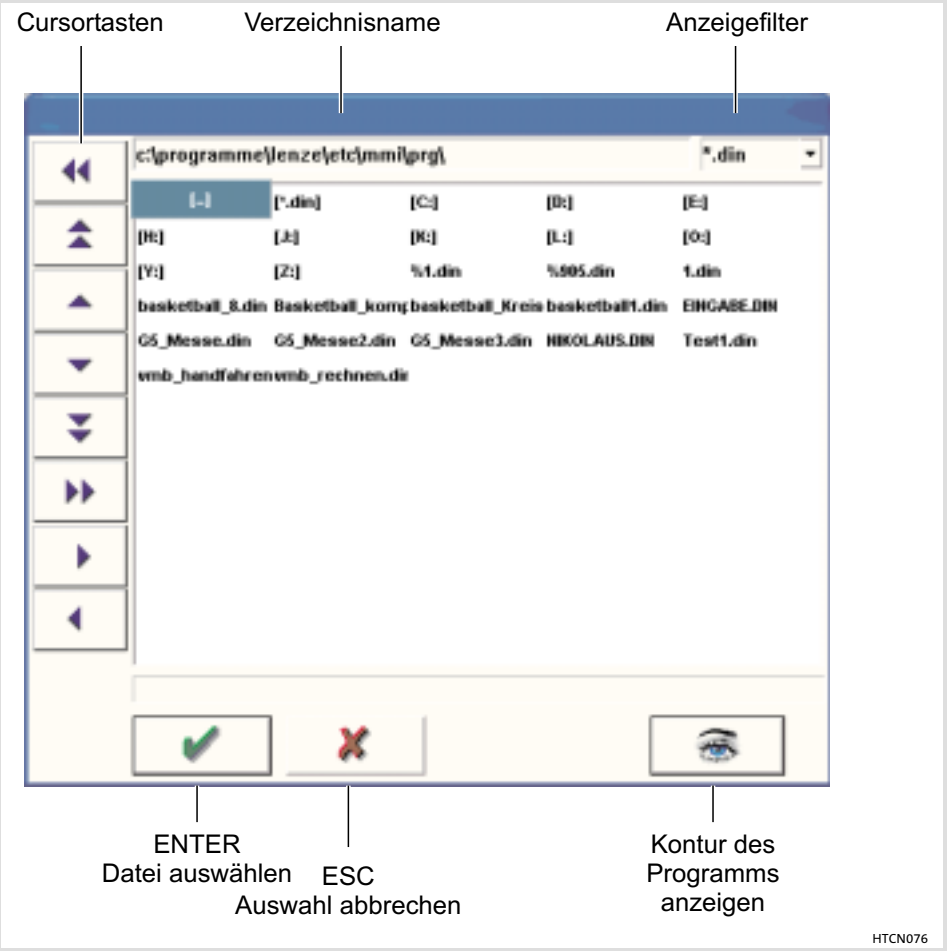
Wir empfehlen, den Editor über eine externe Tastatur zu bedienen. Zur Vereinfachung der Programmerstellung werden CNC-Sätze automatisch in Großbuchstaben gesetzt. Möchten Sie kleine Buchstaben eingeben (z. B. für Achsen u, v, w) drücken Sie die **<Shift>**-Taste während der Eingabe. Innerhalb von Kommentaren schaltet der Editor automatisch auf die übliche Eingabeform um.

Tasten-Funktionen

<Cursor links/rechts>	Cursor um ein Zeichen nach links/rechts bewegen.
<Shift>+<Cursor links/rechts>	Zeichen links/rechts vom Cursor markieren.
<Pos1>	Cursor auf Zeilenanfang setzen.
<Ende>	Cursor auf Zeilenende setzen.
<Ctrl-Pos1>	Cursor auf Dateianfang setzen.
<Ctrl-Ende>	Cursor auf Dateiende setzen
<Bild hoch>	Seitenweise noch oben blättern.
<Bild runter>	Seitenweise noch unten blättern.
<Ctrl>+<C>	Bereich in Zwischenablage kopieren.
<Ctrl>+<X>	Bereich in Zwischenablage verschieben (ausschneiden).
<Ctrl>+<V>	Inhalt der Zwischenablage an Cursorposition einfügen.
<Ctrl>+<Z>	Letzte Änderung einer Zeile zurücknehmen.
<ENTER>	Neue Zeile einfügen.
<Rückschritt>	Zeichen links vom Cursor oder markierten Textblock löschen.
<ENTF	Zeichen rechts von Cursor oder markierten Textblock löschen.

Dialogbox zur Dateiauswahl

Diese Dialogbox dient u. a. zur Auswahl eines Programms für den Editor und für den Transfer an den NC-Rechner.



Allgemeine Funktionen

- ► Einen Dialogbereich (Schaltfläche, Eingabefeld, Auswahlliste usw.) wählen Sie mit der **<TAB>**-Taste an.
- ► Mit den **<Cursor>**-Tasten fahren Sie ein Element im angewählten Dialogbereich an (z. B. eine bestimmte Datei im Dateiauswahlbereich).
- ► Mit **<Enter>**-Taste führen Sie eine Aktion aus. Mit **<ESC>** brechen Sie sie ab.

Datei anwählen

1. Drücken Sie die **<TAB>**-Taste mehrfach, bis das Anzeige- und Eingabefeld des Verzeichnisnamens selektiert ist.
2. Wenn Sie den Dateipfad ändern möchten, drücken Sie **<Enter>** und geben einen neuen Pfad ein. Drücken Sie **<Enter>**, um die Eingabe zu übernehmen oder **<ESC>**, um sie zu verwerfen.
3. Drücken Sie die **<TAB>**-Taste mehrfach, bis das Dateiauswahlfeld selektiert ist.
4. Wählen Sie dort die Datei mit den **<Cursor>**-Tasten an. Drücken Sie **<Enter>**, um die Datei auszuwählen.

Kontur des Programms anzeigen

1. Wählen Sie die Datei, deren Kontur Sie anzeigen möchten (siehe oben).
2. Drücken Sie anschließend mehrfach die **<TAB>**-Taste, bis die Schaltfläche ”Kontur des Programms anzeigen” selektiert ist.
3. Drücken Sie abschließend auf **<Enter>**.
   Die Kontur des selektierten Programms wird angezeigt.
4. Drücken Sie **<ESC>**, um die Anzeige zu schließen.

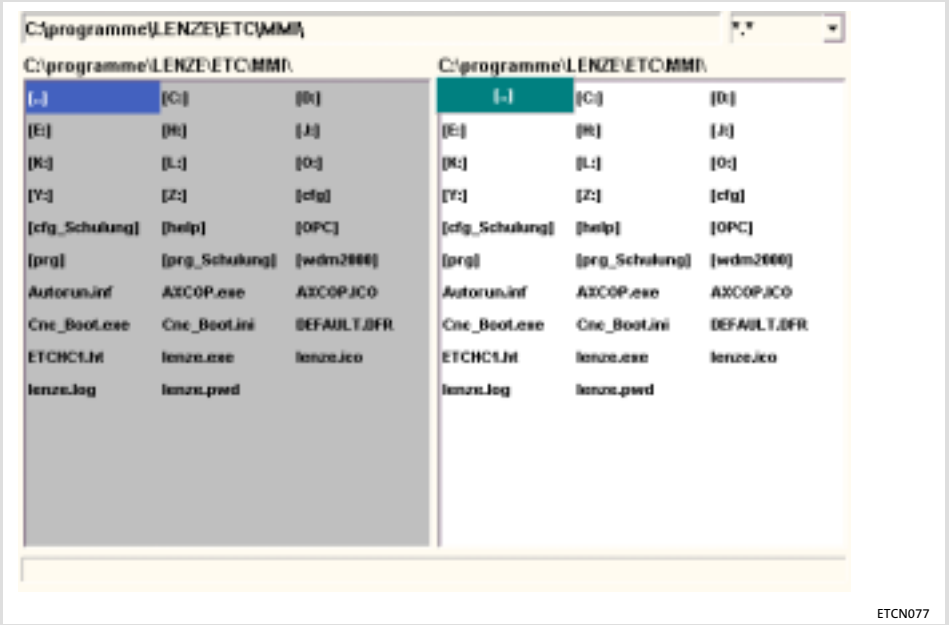
Filter setzen

Mit dem Anzeigefilter kann die Anzeige der Dateien im ”Dateiauswahlfeld” erweitert oder begrenzt werden.

Der Anzeigefilter erscheint zusätzlich zu den Verzeichnissen in eckigen Klammern [*.DIN]. Wird dieser Eintrag ausgewählt, werden alle Dateien des aktuellen Verzeichnisses, die von diesem Filter betroffen sind, zur Steuerung übertragen.

7.6.2 Dateimanager

Der Dateimanager stellt Funktionalitäten zur Verfügung, mit denen Dateien verwaltet werden können. Beim Aufruf des Dateimanagers über den entsprechenden Softkey der Betriebsart ”Programmieren” wird eine neue Softkeyzeile eingeblendet, die alle Funktionen des Dateimanagers bereitstellt.



Im Dateimanager können unabhängig voneinander zwei verschiedene Verzeichnisse im linken und rechten Dateifenster dargestellt werden. Das jeweils aktive Dateifenster ist weiß hinterlegt. Im ”Verzeichniseingabefeld” kann das Verzeichnis des aktiven Dateifensters vorgegeben werden. Das Dateifilterfeld erlaubt, die Dateien in den angegebenen Verzeichnissen für die Anzeige zu filtern. Das Dateifilter wirkt auf beide Dateifenster.

Im oberen Bereich des Dateimanagers sind die verfügbaren Laufwerke und Verzeichnisse aufgelistet. Ein Eintrag mit zwei Punkten ([..]) ist ein Verweis auf das übergeordnete Verzeichnis. Im unteren Bereich werden alle Dateien im aktuellen Verzeichnis gelistet.

Allgemeine Funktionen

- ► Einen Dialogbereich (Schaltfläche, Eingabefeld, Auswahlliste usw.) wählen Sie mit der **<TAB>**-Taste an.
- ► Mit den **<Cursor>**-Tasten fahren Sie ein Element im angewählten Dialogbereich an (z. B. eine bestimmte Datei im Dateiauswahlbereich).
- ► Mit der **<Enter>**-Taste führen Sie eine Aktion aus. Mit **<ESC>** brechen Sie sie ab.

Horizontale Funktionstasten

Kopieren	Markierte Datei(en) kopieren.
Bewegen	Markierte Datei(en) in das andere Verzeichnis verschieben.
Löschen	Markierte Datei(en) löschen.
Umbenennen	Markierte Datei(en) umbenennen.
Drucken	Markierte Datei(en) drucken.
Grafik	Die durch den Cursor ausgewählte Grafik anzeigen. <ESC> schließt die Grafik wieder.
Anzeigen	Die durch den Cursor ausgewählte Datei anzeigen (schreibgeschützt). <ESC> schließt die Grafik wieder.
zurück...	Dateimanager verlassen.

Vertikale Funktionstasten

eine markieren	Die durch den Cursor ausgewählte Datei markieren.
alle markieren	Alle Dateien markieren. Sind bereits Dateien markiert, führt die Funktion eine Umkehrung der Markierung durch.

Hotkeys

c	Kopieren (Copy)
d	Löschen (Delete)
m	Bewegen (Move)
p	Drucken (Print)
r	Umbenennen (Rename)
v	Anzeige (View)
+	Markierung über Filter
-	Markierung über Filter aufheben
*****	Markierung umkehren
.	Punkt, aktuelle Datei markieren/demarkieren

7.6.3 Zyklenprogrammierung

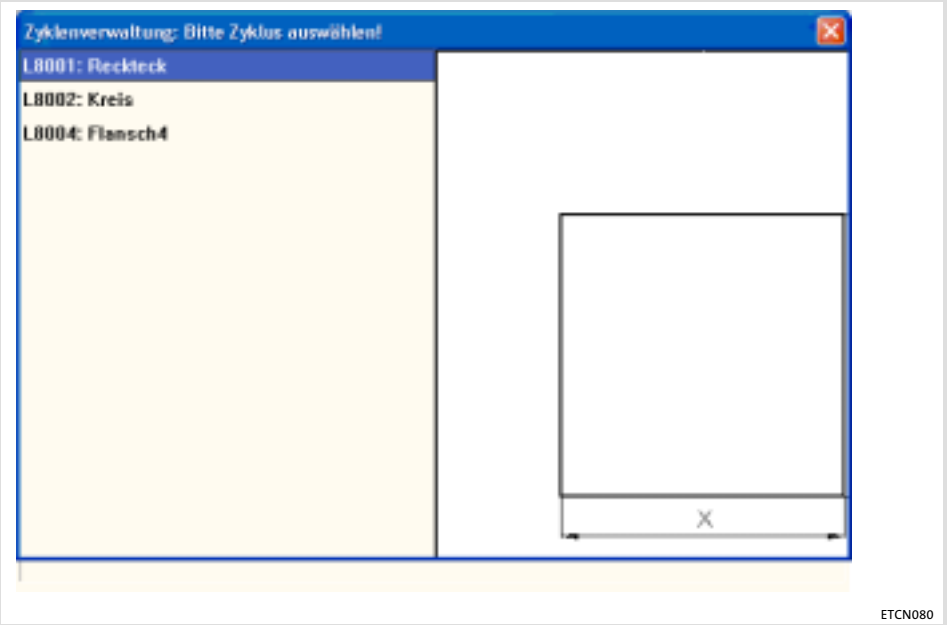
Ein Zyklus ist ein Unterprogramm, dem beim Aufruf Parameter übergeben werden. Bei der Zyklenprogrammierung werden die Parameter vom Benutzer abgefragt und dem DIN-Programm in P-Feldern zur Verfügung gestellt. Vorhandene Zyklen werden in der Konfigurationsdatei definiert. Nähere Informationen zur Zyklenerstellung finden Sie weiter hinten in diesem Kapitel (📖 318).

Zyklen einfügen

Durch Anwahl der Funktion ”Zyklus einfügen” in der Betriebsart ”Programmierung” wird ein Fenster mit der Liste der verfügbaren Zyklen geöffnet. Aus dieser Liste kann der Zyklus ausgewählt werden, der in das angezeigte DIN-Programm eingefügt wird.

Nach Drücken von **<Enter>** werden die Parameter, die zu diesem Zyklus gehören, angezeigt. Mit **<Enter>** gelangt man von Eingabefeld zu Eingabefeld. **<Enter>** beim letzten Feld bewirkt, dass der Zyklus als Unterprogrammaufruf im Editorfeld eingetragen wird.

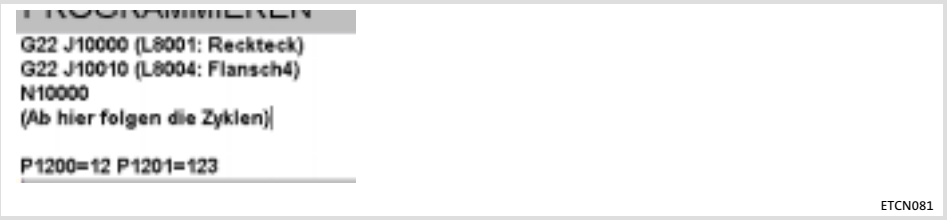
Der eigentliche Programmaufruf und die Übergabe der Parameter wird an das Ende des Programms ab Satznummer 10000 geschrieben.



Beispiel

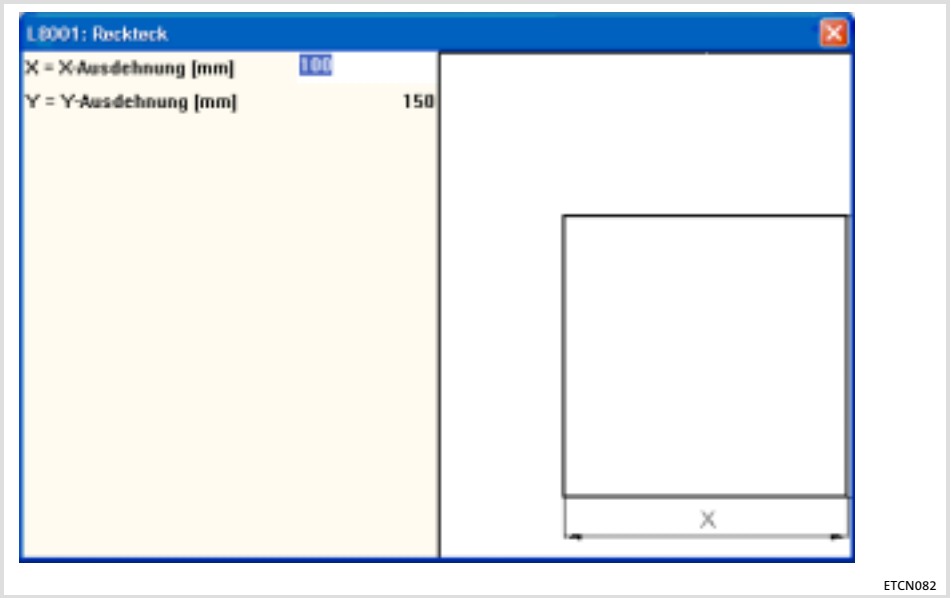
```
%1
G22 J10000 (L3501)
M30
(*** Ab hier folgen die Zyklen
***)
N10000
P1200=100.0000  P1201=100.0000  P1202=20.0000
```

Darstellung des Zyklus im DIN-Programm




```
P1203=20.0000  P1204=20.0000  P1205=1.0000
P1206=0.0000   P1200=90.0000
G22 L3501
G99
```

Zyklen bearbeiten



Nach Auswahl der Funktion bzw. eines Zyklus werden die entsprechenden Einträge im DIN-File gemacht.

Zyklus ändern

Steht der Cursor auf einer Zeile, die mit ”G22 Jxxxxx” beginnt, kann man diesen Zyklus - soweit alle Informationen vorhanden sind - bearbeiten bzw. die Daten ändern.

Zyklus löschen

Steht der Cursor im Editor auf einer Zeile ”G22 Jxxxxx”, kann dieser Zyklus (Aufruf und Parameterteil) aus der DIN-Datei entfernt werden.

7.7 Betriebsart ”Diagnose”

Die Betriebsart ”Diagnose” ist in erster Linie für den Service- und Inbetriebnahmetechniker gedacht.

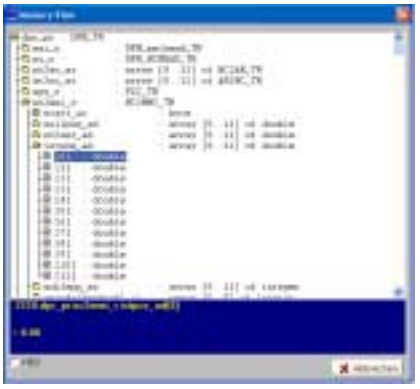
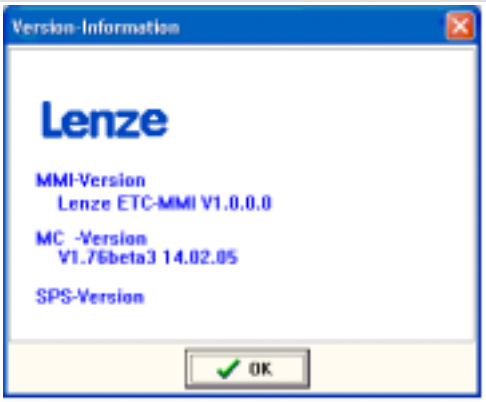
Sie bietet Funktionen zur Unterstützung der Achseinstellung, zur Kontrolle von Ein-/Ausgängen und Parametern, zur Bearbeitung von Maschinenkonstanten sowie zur Kontrolle interner Zustände.

Darüber hinaus ist hier das Fehlerlogbuch zugänglich, in dem alle aufgetretenen Fehler protokolliert sind.

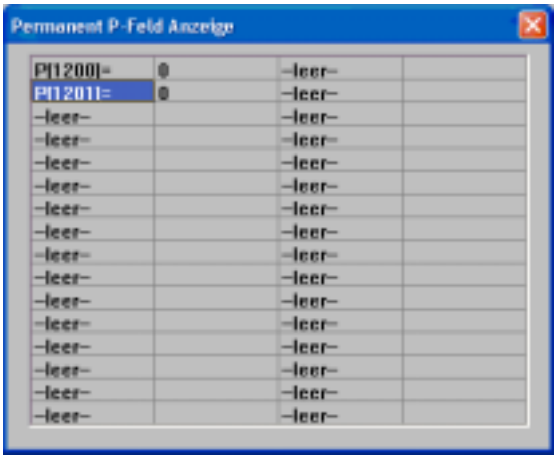
ETCN083

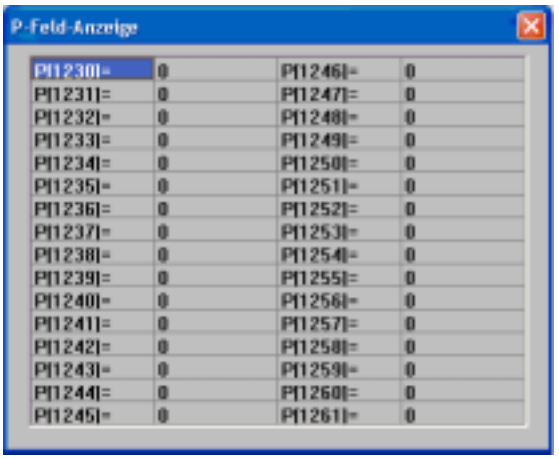
Horizontale Funktionstasten

Achseneinstellung...	Anzeigen zur Unterstützung bei der Achseinstellung aktivieren. Die Handhabung entspricht der Betriebsart EINRICHTEN.
modales Fahren	Die mit der Taste **<Achse>** (oder <Cursor>-Tasten) angewählte Achse wird verfahren, solange die Taste **<Fahren +>/<Fahren ->** betätigt wird. Die Geschwindigkeit, mit der die Achsen verfahren werden, ist durch MK_MODV-MAX vorgegeben, kann aber mit der Taste **<Override +>/<Override ->** beeinflusst werden.
Schlepp-/Mod-/Sollpos.	Umschaltung der Anzeige zwischen: ● Schleppabstand in Eingabeeinheit [mm] ● Soll-Position ● modale Position
Eingabe	Einzelsatz (z. B. G-, S-, T- und M-Funktionen) eingeben. Die Ausführung wird mit **<Enter>** abgeschlossen und damit ausgelöst.
zurück...	Zurück zur vorherigen Ebene.

Diagnosedaten...	
NC-Timing Information	Verschiedene interne steuerungsspezifische Daten anzeigen. Die Funktion ist nur für den internen Gebrauch bzw. für geschultes Service-Personal gedacht. Die Anzeige kann mit der **<ESC>**-Taste gelöscht werden.
Trace	Dialog zur Aktivierung von Ereignissen, die mitgeschrieben werden sollen, öffnen. Aus der Liste der zur Verfügung stehenden Traces können einzelne mit Doppelklick ausgewählt und mit **<Start>** aktiviert werden. Es folgt eine Aufforderung zur Eingabe eines Dateinamens (z. B. \log\ETC_MMI.trc). In diese Datei werden die Ereignisse eingetragen. Die Aktivierungen werden beim Beenden des MMI wieder zurückgesetzt.
Achsen-Information	Parameter der konfigurierten Achsen anzeigen.
Dump	Die Datei ”IBprot.txt” wird im Verzeichnis ”\log” erstellt und angezeigt. Sie enthält eine Reihe von Informationen über das System (u. a. DelphMMI.ini, aktuelle Maschinenkonstanten, aktuelle Werkzeugdaten, Inhalte der P-Felder). Bei Problemen können auf diese Weise relevante Daten zusammengestellt werden.
DPR	Struktur des gesamten DPRs mit den aktuellen Werten anzeigen.  ETCN084
Versionsinformation	Versionsinformationen zur ETC-MMI, Firmware usw.  ETCN085
zurück...	Zurück zur vorherigen Ebene.

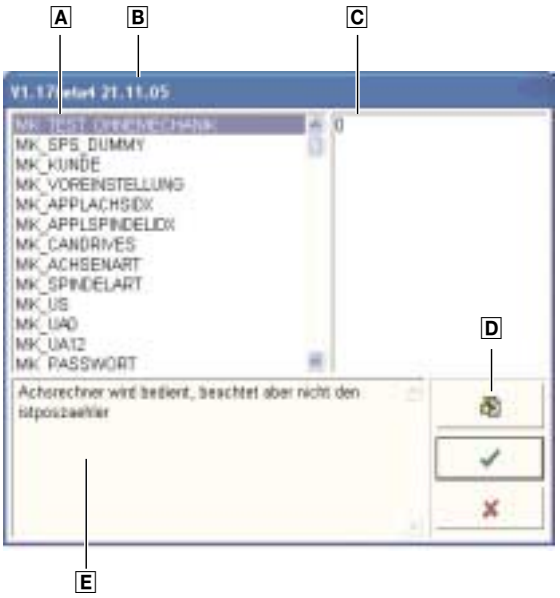
Fehlerlogbuch	Treten bei der Kommunikation von PC, SPS, NC und/ oder bei der Programmierung Fehler auf, werden sie in einer Datei (errorlog.txt) mit Datum und Uhrzeit protokolliert. Mit Hilfe der Funktion ”Fehlerlogbuch” können die Fehlermeldungen angezeigt werden. Die einzelnen Fehler lassen sich mit den Cursor-Tasten durchblättern. Mit **<ESC>** oder **<EXIT>** kann die Anzeige verlassen werden. Fehler, die älter als 30 Tage sind, werden aus dem Logbuch entfernt. Die einzelnen Spalten haben folgende Bedeutungen: ● **Nr:** laufende Nummer ● **Quelle:** Gibt das Modul an, das den Fehler gemeldet hat. ● **Gew:** Gibt die Gewichtung des Fehlers an (z. B. 1 = Information; 4 = Fataler Fehler, der zum Abbruch führt). ● **Fehlernr:** Eindeutige Identifikation des Fehlers ● **Text:** Fehler Text aus einer der Fehlertext Dateien ”*_Fehl.DB” ● **Zusatztext:** Zusätzliche Information des Moduls, die einen wesentlichen Hinweis auf die Ursache der Meldung geben. Die Texte des aktuell markierten Fehlers werden im unteren Teil nochmals angezeigt. Falls z. B. ein Zusatztext innerhalb der Tabelle nicht vollständig sichtbar ist, kann dieser dann hier gelesen werden. Mit dem Filter Source wird ein Modul ausgewählt, dessen Fehler angezeigt werden sollen. Mit *** werden die Meldungen aller Module angezeigt. Darstellung eines Fehlerprotokolls ohne Filter: ETCN086 Darstellung eines Fehlerprotokolls mit Filter: ETCN087
	Bei Betätigung von Filtern werden weitere Möglichkeiten zum Filtern der Meldungen angeboten; zum Beispiel nach einem Zeitraum und nach der Anzahl der anzuzeigenden Meldungen. Mit **<Übernehmen>** werden die Filtereinstellungen übernommen. Mit **<Beenden>** wird der erweiterte Filter wieder ausgeblendet.
Parameterfeld (P-Feld)	Bei allen Eingaben gilt: Die Taste **<Enter>** dient als Bestätigung der Eingabe, die Taste **<ESC>** als Abbruch.

P-Feld lesen	Wert eines Parameterfeldes anzeigen. Die Eingabe der P-Feld-Nummer erfolgt in der Eingabezeile, die anschließende Ausgabe des Inhaltes in der Anzeigezeile. Bei der Anwahl wird automatisch die zuletzt eingegebene Nummer angeboten. Eine andere Nummer kann sofort eingegeben werden, ohne die alten Anzeige zu löschen. ETCN088
P-Feld ändern	Wert des gewünschten Parameterfeldes ändern. Beide Eingaben erfolgen in der Eingabezeile. Für die Eingabe der Ziffern gilt das gleiche wie für ”P-Feld lesen”.
Permanentanzeige	Öffnet ein separates Fenster mit einer Liste von 32 Parametern. Diese werden solange aktuell angezeigt, bis das Dialogfenster mit **<ESC>** wieder geschlossen wird. Eingegebene P-Felder werden in diesem Fenster ständig angezeigt. Die Auswahl der Parameter erfolgt durch Anwahl eines Platzes mit den Cursortasten und Öffnen eines Eingabefensters mit der **<Leertaste>**. Dort wird die gewünschte P-Feld-Nummer eingegeben. Bei Aufruf des Fensters werden automatisch die in der Konfigurationsdatei festgelegten P-Felder angezeigt. Diese Nummern können jederzeit überschrieben werden.  ETCN089
P-Feldanzeige	Öffnet ein separates Fenster mit einer Liste von 32 aufeinanderfolgender Parametern. Der Startparameter kann durch Drücken der **<Leertaste>** oder **<Enter>** eingegeben werden. Wenn das letzte P-Feld angewählt wird, erscheint der P-Feld-Index in der Anzeige. Mit **<Enter>** wird dann dieser Index im ersten Feld angezeigt. Damit kann man gleichsam 31 Felder nach vorn blättern. Wenn auf einem Feld mit Wert des P-Feldes **<Enter>** betätigt wird, kann dieser Wert über einen Eingabeblock geändert werden. Der Dialog kann mit **<ESC>** wieder geschlossen werden.

	 ETCN090
zurück...	Zurück zur vorherigen Ebene.
SPS-Signale	Diese Funktion bietet verschiedene Möglichkeiten, die Kommunikation zwischen MMI und SPS zu überprüfen sowie die Zustände der Ein- und Ausgänge der SPS zu überwachen 📖 237.
MMI <> SPS	Hier werden zwei Fenster eingeblendet, die den Datenaustausch zwischen dem MMI und der SPS (PLC) über den Datenbaustein DB2 zeigen. Im oberen Fenster werden Daten angezeigt, die das MMI-Programm an die SPS sendet. Im unteren Fenster sieht man Daten, die die SPS an das HMI schickt. Mit der **<TAB>**-Taste kann man zwischen den Fenstern umschalten und mit den **<Cursor>**-Tasten die Anzeige weiterrollen. ETCN091
SPS-Freigaben	SPS-Freigaben dienen zur Überprüfung der wichtigsten Schnittstellensignale zwischen SPS und NC.

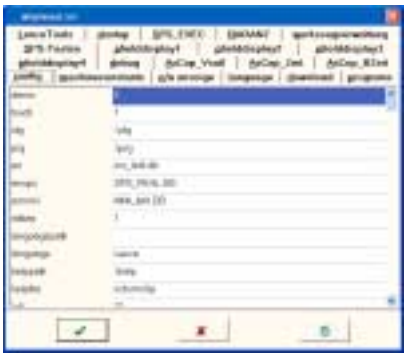
ETCN092

Remote Ein/Aus binär	Hier werden die Ein- und Ausgangszustände der über CAN-Bus angeschlossenen externen E/A-Module in binärer Form angezeigt (0 = Ein-/Ausgang nicht gesetzt). Die Funktion ist nur von Bedeutung, wenn das System mit der entsprechenden Hardware ausgestattet ist.
Karten Ein/Aus ...	
Lokal dig. E/A	In der Sprachdatei kann für jeden Ein- bzw. Ausgang ein Text hinterlegt werden (📖 314). Ist ein Eingang gesetzt, wird er farblich anders dargestellt.
lokal anal. E/A	
Remote E/A	Zeigt eine Liste der konfigurierten CAN-Bus-Module an. Mit den **<Cursor>**-Tasten kann ein Modul ausgewählt werden. Mit **<Enter>** werden die Ein- und Ausgänge dieses Moduls angezeigt.
zurück...	Zurück zur vorherigen Ebene.
zurück...	Zurück zur vorherigen Ebene.
Maschinenkonstanten	Maschinenkonstanten bearbeiten (📖 188).



ETCN093

	Ⓐ: Verfügbare Maschinenkonstanten Ⓑ: Datum und Version der NC Ⓒ: Einstellung der aktuellen Maschinenkonstante Ⓓ: Wert ändern Ⓔ: Beschreibung der Maschinenkonstante aus der ”MK.hlp”. Im Maschinenkonstantenkatalog können einzelne Parameter geändert werden.
MK ändern	Mit ”MK ändern” werden zunächst die aktuellen Maschinenkonstanten vom NC-Rechner geladen und anschließend angezeigt. Nun können sie geändert werden. Mit der **<TAb>**-Taste können Sie die einzelnen Elemente in der Dialogbox anwählen. Befinden Sie sich im Feld, in dem die Maschinenkonstanten aufgelistet sind, können Sie durch Betätigen der **<Leertaste>** die Einstellung(en) für die gewählte Konstante vornehmen. Eingaben sind mit **<Enter>** zu bestätigen. Mit der **<ESC>**-Taste schließen Sie den Dialog zum Ändern von Maschinenkonstanten, ohne dass die Änderungen übernommen werden. Mit **<Enter>** beenden Sie die Bearbeitung. Die Änderungen werden anschließend an den NC gesendet und in der konfigurierten MK-Datei aktualisiert. Somit werden die geänderten Maschinenkonstanten beim nächsten Start wieder geladen.
MK laden	Verfügbare Maschinenkonstanten-Dateien anzeigen. Um eine Datei an die ETCxC zu übertragen, wählen Sie dies mit den **<Cursor>**-Tasten und drücken **<Enter>**. Hinweis: Beim nächsten Start des Steuerungssystems wird wieder die in der INI-Datei eingetragene MK-Datei geladen.
EDS	Beschreibungsdatei des Herstellers eines Can-Moduls laden. Alle definierten Code-Stellen können ausgelesen werden. Die zum Schreiben freigegebenen Code-Stellen können mit neuen Werten beschrieben werden. Details hierzu sind der Modul-Beschreibung des Herstellers der CAN-Module zu entnehmen. Bei Betätigung des Softkeys erscheint zunächst eine Auswahl des Can-Bus (1:I/O-Bus, 2:Antrieb). Anschließend muss die Knotennummer des gewünschten Moduls angegeben werden. Im folgenden Dialog wird die Beschreibungsdatei (*.eds) des Modultyps ausgewählt. Danach kann auf das gewählte Modul zugegriffen werden. ETCN094
zurück...	Zurück zur vorherigen Ebene.

Passwort ändern	Nach Betätigen des Softkeys erscheint ein Dialog. Hier ist zunächst das Passwort zur Berechtigung für Passwortänderungen einzugeben (📖 286.) Anschließend wird eine Auswahlliste angezeigt, in der das zu ändernde Passwort markiert werden muss. Danach wird zweimal das neue Passwort abgefragt, um mögliche Fehleingaben abzufangen. Neue Passworte sind sofort nach der Änderung aktiv. Wenn für bestimmte Betriebsarten keine Passworte gewünscht sind, ist als aktives Passwort ein Leerstring einzugeben.
MMI-Konfiguration	Konfigurationsdatei ”DelphMMI.ini” anzeigen und ändern (📖 320). Hinter jedem Reiter verbirgt sich eine Sektion der Ini-Datei. Mit dem Button rechts unten kann innerhalb einer Sektion ein neuer Eintrag eingefügt werden. Beispielsweise können damit in der Sektion ”programs” Zyklen eingetragen werden, die beim Start geladen werden sollen.  ETCN095

Vertikale Funktionstasten

Start	Startet das ausgewählte Programm bzw. die satzweise Bearbeitung im Einzelsatzbetrieb.
Stop	Beendet unmittelbar die Programmausführung, alle Achsen werden mit den eingestellten Bremsrampen angehalten. Bei einem erneuten Start wird die Bearbeitung von vorn gestartet.
Fahren+	Handverfahrtaste für positive Achsrichtung der ausgewählten Achse.
Fahren-	Handverfahrtaste für negative Achsrichtung der ausgewählten Achse.
Achse	Achse, die verfahren werden soll, auswählen. Alternativ kann die Achse mit den **<Cursor>**-Tasten angewählt werden.
Override+	Achs- bzw. Bahngeschwindigkeit der angewählten Achse vergrößern (in Prozent von MK_VMAX). Der Wert wird oberhalb von 10 % in 10 %-Schritten verändert und unterhalb von 10 % in 1 %-Schritten.
Override-	Achs- bzw. Bahngeschwindigkeit der angewählten Achse verkleinern (in Prozent von MK_VMAX). Der Wert wird oberhalb von 10 % in 10 %-Schritten verändert und unterhalb von 10 % in 1 %-Schritten.

7.8 Anhang

7.8.1 Sprachdatei (SPRACHE.TXT)

Die Sprachdatei (z. B. SPRACHE.TXT) enthält alle Anzeigen- und Meldungstexte ausschließlich der Fehlermeldungstexte, die in gesonderten Dateien hinterlegt sind. Die Datei ist im Windows-üblichen ANSI-Zeichensatz erstellt und kann mit jedem beliebigen Editor (z. B. dem Editor in der Betriebsart PROGRAMMIEREN der MMI-Software) editiert werden.

Der Anfang einer Textzeile wird immer durch eine Nummer gekennzeichnet. Die Nummer dient der MMI-Software als Auswahlkriterium. Steht am Anfang einer Zeile keine Nummer, so wird der Text mit Zeilenumbruch zu dem voranstehenden Text zugeschlagen. Teilweise enthalten die Texte noch Zusatzinformationen, die nicht direkt so zur Anzeige kommen.

- Die Softkeybeschriftungen beinhalten neben dem Text auch noch die Informationen über die Farbe des Softkeys (@C), ein optionales Bitmap, dessen Ausrichtung und eventuelle Animation (@B), die Textausrichtung, den zu verwendenden Font, dessen Größe und Farbe (@F). Zum Beispiel:

  00000000,”Menu@C12632256@BFLDRMANY,4,4,2,1,1,0,0@FArial,8,1,0,3,6”

- Ist einem Text eine Tilde (”~”) angehängt, wird eine Anweisung ”%Axx~” im entsprechenden Text durch den Kennbuchstaben der mit ”xx” ausgewiesenen Achse ersetzt. Der Kennbuchstabe ergibt sich hierbei durch die aktuelle Maschinenkonfiguration. Der Index der Achse wird durch die Angabe ”xx” festgelegt. Bei einer Maschinenkonfiguration ”XZC” wird der Text ”Das ist die %A2 Achse~” als ”Das ist die C Achse” zur Anzeige gebracht.

Die Nummern -1 ... 00001323 werden zur Beschriftung der Funktionstasten (Softkeys) verwendet. Nummern kleiner als 3000 dürfen nur für Funktionstasten verwendet werden, nicht für sonstige Anzeigen oder dergleichen.

Beispiele aus der Sprachdatei

```
-0000001,”@C12632256@FArial,8,1,0,3,6” <- leere Funktionstaste
00000000,”Menu@C12632256@BFLDRMANY,4,4,2,1,1,0,0@FArial,8,1,0,3,6”
00000001,”Einrichten@C12632256@FArial,8,1,0,0,15359”
   .
   .
   .
00001312,”zurück ...@C12632256@FArial,8,1,0,3,6”
00001323,”Zyklus löschen@C12632256@FArial,8,1,0,3,6”
.
.
00002800,” 1.te SPS-Taste@C12632256@FArial,8,1,0,3,6” <- Beschriftung für optionale SPS-Tasten
00002801,” 2.te SPS-Taste@C12632256@FArial,8,1,0,3,6” (siehe Sektion [SPS-Tasten] in DELPHMMI.INI)
.
.
00002926,”127.te SPS-Taste@C12632256@FArial,8,1,0,3,6”
00002927,”128.te SPS-Taste@C12632256@FArial,8,1,0,3,6”
```

Ab Nr. 3000 folgen allgemeine Texte, sowohl für zusätzliche Tasten als auch für Hinweise, Eingaben usw.

```
00003000,"Ihr Passwort bitte :"
00003001,"Passwort Einrichten"
.
.
.
01000007,"mk"
01000008,"Datei wurde geändert.."
"Änderung speichern ?"
01000009,"Zyklusdatei können nicht als Auftrag geladen werden"
01000010,"Die Wert der Eingabe ist unzulässig"
"Für diesen Parameter ist nur ein Wert zwischen %f und %f zugelassen"
01000011,"Eingabefehler"
```

Die Texte Nr. 01010000 ... 01010064 werden bei der Funktion "SPS-Signale -> Lokal Ein/Aus Text" in der Betriebsart DIAGNOSE verwendet. Man kann dadurch den Ein- und Ausgängen der Steuerung eindeutige Namen bzw. Bezeichnungen geben (E 0.0 ... E 3.7 sind die Eingänge).

```
01010000,"Basis IO's"
01010001," E 0.0 hier ist e0"
01010002," E 0.1"
.
01010031," E 3.6"
01010032," E 3.7"
01010033," A 0.0"
01010034," A 0.1"
.
.
01010063," A 3.6"
01010064," A 3.7"
```

Die Texte Nr. 01010100 ... 01010164 werden bei der Funktion "SPS-Signale -> Remote Ein/Aus Text" in der Betriebsart DIAGNOSE verwendet. Man kann dadurch den externen CAN-Bus E/A-Modulen eindeutige Namen bzw. Bezeichnungen geben (E 0.0 ... E 3.7 sind die Eingänge).

```
01010100,"Remote IO's (CAN-Bus)"
01010101," E 0.0"
01010102," E 0.1"
.
.
01010131," E 3.6"
01010132," E 3.7"
01010133," A 0.0"
01010134," A 0.1"
.
.
01010163," A 3.6"
01010164," A 3.7"
```

Ab Nummer 01020000 (bis 01020255) folgen SPS-Meldungen. Diese werden in der MMI-Software in den Betriebsarten EINRICHTEN, AUTOMATIK und DIAGNOSE im Fenster "SPS-Meldungen" angezeigt. Die Meldungen müssen fortlaufend angegeben werden. Das heißt, es dürfen keine Lücken in der Nummerierung vorhanden sein. Die SPS kann die Meldungen ausgeben, in dem sie auf den Datenbaustein DB2 der Dual-Port-Ram-Schnittstelle die Bits DB 80.0 bis DB 95.15 setzt bzw. löscht. In der MMI-Software wird eine Meldung ausgegeben, wenn das entsprechende Bit gesetzt ist. Wird ein "!" dem Text vorangestellt, dann ändert das SPS-Meldungsfenster seine Hintergrundfarbe, wenn diese Meldung ausgegeben wird.

Sinnvoll ist es, diese Texte in einer gesonderten Sprachdatei zu speichern (s. Kapitel Konfigurationsdatei: Language). Die SPS-Texte können dann unabhängig von den MMI-Texten gepflegt werden.

```
01020000,""
01020001,"Bit01"
01020002,"Bit02"
01020003,"!Bit03"
.
.
.
01020027,"Bit27"
01020028,"Bit28"
01020029,"Bit29"
```

Ab Nr. 1020300 folgen wieder allgemeine Texte, Fehlermeldungen usw.

```
01020300,""
02000000,"Problem bei der Maschinenkonstantenübertragung"
02000001,"Zuviele Dateien gleichzeitig geöffnet"
   .
   .
   .
20000005,"Schritt fahren   (Schrittweite Achse +/-)"
20000006,"Modal fahren     (Achse +/-)"
```

Kartenspezifische E/A-Anzeige

Im Pfad Diagnose\SPS Signale kann über den Softkey "F6" die kartenspezifische Anzeige der digitalen und analogen Ein-/Ausgänge angewählt werden. Die Ein-/Ausgangsmodule sind in 3 Kategorien gegliedert. Mit den folgenden Softkeys werden für den angewählten Typ die angeschlossenen Module in einer Listbox angezeigt.

Softkey "F1": Es werden die lokalen digitalen Ein-/Ausgabemodule angezeigt.

Softkey "F2": Es werden die lokalen analogen Ein-/Ausgabemodule angezeigt.

Softkey "F3": Es werden die remote Ein-/Ausgabemodule, die über die CAN-Bus Schnittstelle angesprochen werden, angezeigt.

Mit "ENTER" oder mit "Doppel Click" wird eine Karte ausgewählt. Die E/A-Anzeige wird entsprechend der Kartenkonfiguration aufgebaut. Die Anzeige ist in Zeilen und Spalten aufgebaut.

Digitale Eingänge: Der Ein-Zustand wird mit einem Farbumschlag gekennzeichnet.

Digitaler Ausgang: Über die Cursortasten kann ein digitaler Ausgang angewählt werden. Mit der ENTER-Taste wird der Ausgang entweder ein- oder ausgeschaltet.

Analoge Ein-/Ausgänge: Diese Anzeigen werden zweispaltig dargestellt. In einer Spalte erscheint die Beschreibung des Ein- bzw. Ausgangs und in der Spalte daneben erscheint der zugeordnete Wert. Der Wert wird als nicht normierter und vorzeichenloser 16 Bit Wert dargestellt.

Die Maske kann mit der ESC-Taste oder durch Anwahl der o. g. Softkeys verlassen werden.

Textzuordnung der Ein-/Ausgänge in der Sprachdatei

Die Textzuordnung der einzelnen Ein- und Ausgänge erfolgt über Textnummern. Die Textnummern sind nach folgendem Schlüssel aufgebaut. Jedes Zeichen entspricht einer Ziffer.

Schlüssel	TKKKXNNN (8stellige Nummer)
T	Typ 0: Lokales digitales E/A-Modul (EC-IO) 1: lokales analoges E/A-Modul (EC-ADA, EC-ADC) 2: CAN-E/A-Modul (SLIO, CANOpen)
KKK	Knotennummer 001 ... 127 bei CAN-Bus Modulen 001 ... 008 bei lokalen Modulen
X	Typ des E/A-Signals 0: unbenutzt 1: digitaler Eingang 2: digitaler Ausgang 3: analoger Eingang 4: analoger Ausgang
NNN	Nummer des E/A-Signals 000 ... 063 für CAN-Bus digitale E/A-Signale 000 ... 007 für CAN-Bus analoge E/A-Signale 000 ... 023 für lokale Eingänge 000 ... 015 für lokale Ausgänge

Auf diese Nummer wird eine Basisnummer von 30000000 addiert. Diese Basisnummer lässt sich über einen Parameter in der Konfigurationsdatei ändern. Die Texte innerhalb der Sprachdatei können damit verschoben werden.

Beispiel

Text in der Sprachdatei: 30011000," E 0.0 hier ist e0"

30000000	T = 0	Ab dieser Nummer beginnen die E/A-Signal-Texte.
1	KKK = 1	Kartennummer 1. Lokale Karte
1	X = 1	digitaler Eingang
000	NNN = 000	Nummer des digitalen Eingangs

Softkey-Zuordnung

Die folgenden Textnummern sind den Softkeys zugeordnet:

```
00000037,"Karten Ein/Aus ...@C12632256@FArial,8,1,0,3,6"
00001384,"Lokale dig. E/A@C12632256@FArial,8,1,0,3,6"
00001385,"Lokale anal. E/A@C12632256@FArial,8,1,0,3,6"
00001386,"Remote E/A@C12632256@FArial,8,1,0,3,6"
```

Überschrift-Zuordnung

Die folgenden Textnummern werden für Überschriften und Labels verwendet:

```
00011200,"IO Anzeige :"
00011201,"digital In"
00011202,"digital Out"
00011203,"analog In"
00011204,"analog Out"
00011205,"EC - IO"
00011206,"EC - ADA"
00011207,"CAN"
00011208,"Wert"
```

Einstellungen in der Konfigurationsdatei "delphmmi.ini"

In der Konfigurationsdatei "delphmmi.ini" können im Abschnitt "e/a anzeige" die folgenden Parameter eingestellt werden:

textbasisnr: Basisnummer, ab der die E/A-Signal Texte beginnen.

separate_karten_texte: Darüber lässt sich einstellen, ob für jedes E/A-Modul separate Texte eingegeben werden sollen.

1: Für jedes Modul separate Texte
0: Für alle Module werden die Texte von Modul 1 genommen

Beispiel

[e/a anzeige]

textbasisnr=30000000

seperate_karten_texte=1

7.8.2 Zyklenprogrammierung

Für jeden Zyklus ist ein Eintrag zu erstellen mit Angabe der Zyklennummer in Form eines Sektionsnamens (Lxxxx, wobei 8000 ≥ xxxx ≥ 9999) und einem Klartextnamen. Dieser Name wird in der Betriebsart "Programmieren" bei Auswahl des Zyklus angezeigt.

In den Sektionen Lxxxx werden die anzubietenden Eingabeparameter festgelegt. Darüber hinaus kann eine bmp-Datei angegeben werden, welches bei Auswahl des Zyklus angezeigt wird.

Ein Eingabeparameter wird definiert durch eine P-Feld-Nummer, in der der Eingabewert an den Zyklus übergeben wird.

- Defaultwert
- Minimalwert (optional)
- Maximalwert (optional)
- Kennung für zwingende Eingabe (optional)
- Anzeigetext

enty(n) = [pfeldnr.] = x.xx; [Min]; [Max]; [Flag]; [Text]

Nachfolgend einige Beispiele für Zyklen und deren Parameter:

Sektion	Term	Beschreibung
[Zyklen]		
	entry1 = L8000; Schleifen	entry1: wird zur Identifizierung benötigt. L8000: Sektionsnamen, unter dem der Zyklus und die Zyklusnummer beschrieben wird. Schleifen: Text zur Anzeige und zur Auswahl
	entry2 = L8001; Sägen	
	entry3 = L8002; Nähen	
	entry4 = L8003; frei	
	entry5 = L8004; frei	
[L8000]		

Sektion	Term	Beschreibung
	entry1 = P1200 = 1.23; oben	entry1: Zur Identifizierung P1200: Definition des entsprechenden P-Feldes 1.23: Wert, der ins P-Feld gesetzt wird Oben: Text zur Anzeige und Auswahl
	entry2 = P1201= 2.34; unten	
	entry3 = P1202= 3.45; Mitte	
	entry4 = c:\stdmmi\cnc20.bmp	Bilddatei, die angezeigt werden kann
[L8001]		
	entry1 = P1200= 1.23; Min; Max; Flag; oben	P1200=: Definition des entsprechenden P-Feldes 1.23;: Wert, der ins P-Feld gesetzt wird Min;: Minimaler Wert, der eingegeben werden darf Max;: Maximaler Wert, der eingeben werden darf Flag = 1: Dann muss der Wert eingegeben werden oben: Text zur Anzeige und Auswahl
	entry2 = P1201= 2.34;0;3.14;; unten	Bei diesem Beispiel prüft die Software eine Eingabe auf Min/Max ab. Erfolgt keine Eingabe, wird das Feld nicht im DIN-File eingetragen.
	entry3 = P1202= 3.45; Mitte	
	entry4 =	
[L8002]		
	Entry0 = c:\programme\Lenze\ETC\mmi\Lenze.bmp, 25	Name einer Bilddatei, die angezeigt werden kann. Die 25 sagt aus, dass der Grafik nicht die halbe Bildschirmbreite, sondern nur 25% davon zur Verfügung steht. Der Wert kann eingegeben werden im Bereich von 0 ... 100%.

Sprachunabhängigkeit

Damit eine Sprachunabhängigkeit gewährleistet ist, können die Anzeigetexte auch aus einer anderen Datei als delphmmi.ini geholt werden. Als Quelldatei dient dabei die Datei, die in der INI-Datei (z. B. delphmmi.ini) bei der Sektion [config] unter dem Eintrag [language] angegeben wurde (ohne die Dateierweiterung ".txt").

Beispiel

Sektion	Term	Beschreibung
[Zyklen]		
	entry1 = L8000; #4100	Hier wird der Text mit der Nr. 4100 aus sprache.txt angezeigt.
[L8000]		
	entry1 = P1200= 1.23; Min; Max; Flag; #4151	Hier wird der Text mit der Nr. 4151 aus sprache.txt angezeigt.

7.8.3 Konfigurationsdatei (DELPHMMI.INI)

Die Konfigurationsdatei (standardmäßig delphmmi.ini) enthält Einstellungen, die zum Betrieb des IPC und des NC-Rechners notwendig sind und sollte nur von geschultem Personal geändert werden.

Definitionen

Als Sektion bezeichnet man die Ausdrücke, die in ”[ ]” eingeschlossen sind (Beispiel [config]). Als Term bezeichnet man Ausdrücke, die nach einer Sektion folgen (Beispiel: cfg=c:\programme\Lenze\ETC\mmi\cfg).

Beachten Sie bitte, dass alle Pfadangaben innerhalb der Konfigurationsdatei wahlweise auch relativ zum Pfad des ausführbaren Programms gemacht werden können. Das heißt, wenn das ausführbare Programm auf ”c:\programme\Lenze\ETC\mmi” abgelegt ist, so erreichen Sie mit der Angabe ”.\cfg” das gleiche Unterverzeichnis wie mit der Angabe ”c:\programme\Lenze\ETC\mmi\cfg”.

Inhalt

Sektion	Term	Beschreibung
[config]		
	demo=1	1 -> Betrieb ohne Steuerung (Demo Installation) 0 -> Betrieb mit Steuerung
	mmiTrace=0	Tracemitschriebe der Bediensoftware mitloggen. Wird vom Hersteller zur Fehlersuche verwendet.
	axformat=%.3f	Formatierungsanweisung zur Darstellung der Achsanzeige, um mehr oder weniger Nachkommastellen anzuzeigen.
	nc=ETC_Kunde	Kennung, welche Steuerung angesprochen werden soll. Grundeinstellung erfolgt automatisch bei der Installation (s. ETC-MMI Gateway). Der Steuerungsname kann in MMIGTWAY.EXE eingestellt werden.
	touch=1	Touch-Screen-Anzeige aktivieren
	cfg=.\cfg	Name des Verzeichnisses, in dem sich alle Dateien zur Konfiguration befinden. In dem hier angegebenen Verzeichnis wird u. a. auch die Fehlerdatei namens **errorlog.txt** angelegt.
	prg=.\prg	Name des Verzeichnisses, in dem sich DIN-Programme befinden.
	err=ncr_fehl.db	In dieser ASCII-Datei (Verzeichnis wie unter Term=cfg angegeben) stehen Fehlertexte, die den NC-Rechner betreffen.
	errsps=sps_fehl.db	In dieser ASCII-Datei (Verzeichnis wie unter Term=cfg angegeben) stehen Fehlertexte, die die SPS produziert.
	errmmi=mmi_fehl.db	In dieser ASCII-Datei (Verzeichnis wie unter Term=cfg angegeben) stehen Fehlertexte, die den IPC betreffen.
	initbta=2	Der hier angegebene Wert bestimmt die Betriebsart des IPC, die nach erfolgreichem Programmstart angezeigt wird. 1: Einrichten 2: Automatik 3: Programmieren 4: Diagnose

Sektion	Term	Beschreibung
	LanguagePath	Pfad, in dem die Sprach-Dateien gesucht werden.
	language=sprache	Hier kann der Name einer ASCII-Datei angegeben werden (Namenserweiterung ".txt" wird vorausgesetzt). In dieser Datei liegen allgemeine Anzeigetexte, weitere Fehlermeldungen, die Funktionstastenbelegung usw., die für den Betrieb der MMI-Software notwendig sind (s. auch "Sprachdatei"). Falls hier kein Eintrag erfolgt, kann in einer Section Language eine Liste von Dateien eingetragen werden. Die vergebenen Textnummern, dürfen dabei nicht doppelt vergeben werden. Anmerkung: Alle ASCII-Dateien können mit jedem beliebigen Editor (auch mit der ETC-MMI-Software) bearbeitet werden. Alle Texte in den Dateien können somit auf eigene Bedürfnisse (z. B. in Bezug auf eine andere Landessprache) angepasst werden.
	left=0	Linke Fensterposition, wenn doscal <> 0
	top=0	Obere Fensterposition, wenn doscal <> 0
	doscal=0	Doscal = 0 bedeutet, dass die Software sich optisch so präsentiert, wie sie programmiert wurde. Mit doscal <> 0 kann man die Fenstergröße der MMI-Software verändern. Doscal = 1 bedeutet, dass der komplette Bildschirm für die Applikation zur Verfügung gestellt wird. Die linke obere Ecke wird durch die Terme left und top definiert. Doscal = 2 bis doscal = 1000 bedeutet, dass das Fenster in seiner Größe im Bereich von 0,2% bis 100% der kompletten Bildschirmgröße variiert werden kann. Achtung: Man kann das Fenster so klein machen, dass man es nicht mehr sieht.
	watchdog=0	Watchdog <> 0 gesetzt; bietet die Möglichkeit einer zyklischen Überprüfung, ob der IPC mit der SPS kommuniziert, d. h. ob die Verbindung noch besteht. Voraussetzung für die Funktionsfähigkeit ist, dass das laufende SPS-Programm seinerseits die Watchdog-Funktion unterstützt. Da die Funktion watchdog vom SPS-Beispielprogramm spsdummy.loc nicht unterstützt wird, wird sie durch die Anweisung = 0 inaktiviert.
	QuitMode=0	Ein Wert von 1 fährt Windows nach dem Beenden des Programms herunter. Bei Null wird Windows nicht beendet.
	DisplayFontSize=6	
	HelpPath	Gibt den Pfad an, in dem die Hilfedateien gesucht werden.

Sektion	Term	Beschreibung
	helpfile=c:\programme\Lenze\ETC\mmi\ncform31.hlp	Dieser Term bietet die Möglichkeit, dass die MMI-Software die Einblendung von Online-Hilfen vornimmt. Gibt man den Dateinamen einer Windows-Hilfe-Datei an, erfolgt die Online-Hilfe im Windows-Format. Wird nur ein Verzeichnis (z. B. c:\programme\Lenze\ETC\mmi\help) angegeben, zeigt die Standard-Software Hilfetexte in Form von ASCII-Dateien an. Diese ASCII-Dateien müssen sich in dem angegebenen Verzeichnis befinden. Innerhalb der MMI-Software wird die funktionsbezogene Hilfe durch das gleichzeitige Drücken der "Control"-Taste in Verbindung mit einer Funktionstaste angefordert. Die Namenkonvention für die Datei, die angezeigt wird, ist hierbei <Nummer-der-Funktion>.txt.
	NewTast=0	Um die Taste des IPC mit der second Funktion ohne Tastatur-Treiber zu aktiveren, muss NewTast auf 1 gesetzt werden. Standardmäßig ist der Tastatur-Treiber installiert und NewTast = 0.
	Scanzposfile= testaufz.txt	Defaultname der ASCII Datei, in der die Positionswerte im ASCII Format abgelegt werden.
	Zposkorrfile= testaufz.kor	3D Korrekturfile wird zur Steuerung übertragen.
	saveanddownload=0	Wird SaveAndDownload auf 1 gesetzt, wird im Programmieren nach dem Speichern eines Programms dieses gleich zur Steuerung übertragen.
	startcycle=12345	Dieser Term findet bei der Programmierung von Zyklen Verwendung (s. auch "Zyklusprogrammierung").
[machineconstants]		
	mk=default.mk	Diese Datei enthält Maschinenkonstanten, die an den NC-Rechner gesendet werden. Die Übertragung erfolgt nur, wenn die HMI-Software gestartet wird und auf dem NC-Rechner noch kein Betriebssystem läuft. Gegebenenfalls muss der NC-Rechner zurückgesetzt werden, damit die Maschinenkonstanten übertragen werden können.
	mkhlp=mk.hlp	Diese Datei enthält Hilfetexte, die angezeigt werden, wenn mit dem MMI Maschinenkonstanten geändert werden (Betriebsart Diagnose – Funktion Maschinenkonst. -> MK ändern).
[download]		
	entry1=ETCxC; .\cfg\ETCxC.rsc;	Betriebssystem für den NC-Rechner: Das Betriebssystem wird nur nach einem Reset des NC-Rechners übertragen.
	entry2=ncr; .\cfg\spsdummy.prg;	SPS-Programm: Das SPS-Programm wird nur nach einem Reset des NC-Rechners übertragen. Anmerkung: An dieser Stelle muss nach jeder Installation manuell eingegriffen werden. Sie müssen hier die von Ihnen verwendete SPS-Datei eintragen.

Sektion	Term	Beschreibung
[programs]		
	entry0=d:\demo\std\ cfg\9000.zyk	DIN-Programm: Sollen mehrere DIN-Programme geladen werden, sind mehrere Terme (entry1=..., entry2=... usw.) anzugeben. Das Programm wird nur nach einem Reset des NC-Rechners übertragen.
[startup]		In der Sektion [startup] werden zusätzliche Abläufe während des Hochlaufens der HMI-Software festgelegt.
	dinLoad=1	Dieser boolsche Wert definiert (=1), ob nach dem vollständigen(!) Download des NCR das zuletzt benutzte Automatik-DIN-Programm wieder in die Steuerung geladen und für den Programmstart vorbereitet werden soll.
	dinFile=c:\prog\beispiel.din	Der Name des zuletzt geladenen Automatik-DIN-Programms. Dieser Eintrag wird durch das HMI aktualisiert.
	StartPrg=	Nummer des Programms (1 ... 7999) eingeben, das immer gestartet werden soll. Ist keine Nummer oder Null eingetragen, wird das zuletzt geladene Programm gestartet.
	AxeKorr=	Beim Programmstart wird eine Spindelsteigungskorrekturdatei übertragen.
	logo= c:\bilder\logo.bmp	Spezifiziert ein Bitmap, welches zur Anzeige kommen soll. Ist keine Datei angegeben, erscheint das default Bitmap der Applikation.
	wait=1	Der integer Wert (in Sekunden) legt fest, wie lange die Anwendung während der Anzeige des Logos warten soll (>0), bevor sie mit dem Hochlauf des Systems fortfährt. Der Wert wird auf die Vorgabe unter dem Term ”time” begrenzt.
	time=5	Gibt die Zeitspanne (in Sekunden) an, die das Logo angezeigt wird. Ist ein Wert 0 eingetragen, so wird die Anzeige des Logos unterdrückt.
	Top=10	Definiert die Y-Koordinate der linken oberen Ecke des Bildes bei der Anzeige.
	left=10	Definiert die X-Koordinate der linken oberen Ecke des Bildes bei der Anzeige.
[e/a anzeige]		
	Textbasisnr=30000000	Basisnummer in der Sprache.txt, ab der die E/A-Signal-Texte beginnen.
	Separate_karten_texte=1	1: für jedes Modul separate Texte 0: für alle Module werden die Texte von Modul 1 genommen
[language]		
	Text0=Sprache.txt Text1=SPS.txt	Falls der Eintrag Language in Sektion [Config] leer bleibt, wird auf die Sektion [language] zurückgegriffen. Dabei werden Texte mehrerer Dateien definiert. Es können z. B. Texte vom MMI Entwickler gepflegt werden, andere vom SPS Programmierer und wieder andere vom Endkunden. Es ist dabei auf eine strenge Vergabe der Textnummern zu achten.

Sektion	Term	Beschreibung
[SPS_EXEC]		Aufruf einer unter Windows ausführbaren Datei durch eine SPS-Nachricht: Entry=EXE-Datei;Parameter;[MAX|MIN|NORM] EXE-Datei: ausführbare Windowsdatei Parameter: Übergabeparameter an EXE-Datei MAX: maximales Fenster MIN: minimales Fenster NORM: normales Fenster
entry0=C:\WinNT\Notepad entry1=C:\WinNT\Notepad;Readme.txt entry2=C:\WinNT\Notepad;Readme.txt;Max entry3=C:\WinNT\Notepad;;Min		Beispiele: Siehe auch Abschnitt SPS_EXEC
[LenzeTools]		
F1=C:\programme\lenze\GDC_4_70\bin\gdc32.exe F2=C:\programme\lenze\ETC\MMI\axcop.exe F3=C:\programme\lenze\ETC\CoDeSys\CoDeSys.exe F4=C:\programme\lenze\GDO_1_20\osci.exe F5=c:\programme\lenze\etc\mmi\etchc1.ht		Mit KeyCode (F12-Diagnose, F1-Achseinstellungen, F2-Lenze-Tools) werden die Tools Tasten aktiviert. Die Texte für die Tastenbeschriftung sind in der Sprachdatei ab Eintrag 1501 für F1 vorzunehmen.
[BARANZ]		Konfiguration der Balkenanzeige
ANZ1=0;800;10802;1;P30 ANZ2=0;800;10902;2;P31 ANZ3=0;3000;10920;3;4711;DOUBLE Anz4=0;4000;10930;4;D10,WORD		Siehe auch Abschnitt BARANZ im Anschluss an dieser Tabelle.
[pfielddisplay1] [pfielddisplay2] [pfielddisplay3] [pfielddisplay4]		Die Angaben in diesen Sektionen werden beim Betriebsartenwechsel durch das HMI ausgewertet, aber standardmäßig nicht benutzt. Jede Sektion beschreibt jeweils für eine Betriebsart die Indizes der P-Felder, die durch den NCR zyklisch im Bereich der "P-Feld-Permanent-Anzeige" des DPR=s aktualisiert werden sollen. Die Zuordnung der Sektionen zu den Betriebsarten ist hierbei fest: 1: Einrichten 2: Automatik 3: Programmieren 4: Diagnose
	P0=1200 . . P31=-1	Beispiel: Der Inhalt des P-Feldes 1200 wird im DPR unter nc2mmi_r.pfanz_r.val_ad[0] zyklisch aktualisiert. Bei -1 erfolgt keine Aktualisierung auf diesem Index.
[Zyklen]		(s. auch: "Zyklenprogrammierung")
	entry0 = L8000; Schleifen	L8000 verweist auf eine Sektion, unter der der Zyklus näher beschrieben wird. Schleifen ist ein Text, der zur Zyklusauswahl benutzt wird.
[L8000]		(s. auch: "Zyklenprogrammierung")
	entry1 = P1200= 1.23; oben entry2 = P1201= 2.34; Mitte entry3 = P1202= 3.45; unten	Die einzelnen Einträge beschreiben den Zyklus.
[WerkzeugVerwaltung]		Festlegung der internen Werkzeugverwaltung (s. Dokumentation zum Standard HMI).

Sektion	Term	Beschreibung
	Wsk=default.wsk	Name und Pfadangabe einer Datei mit Werkstückkorrekturdaten. Diese Datei wird nach einem vollständigen(!) Download zur Voreinstellung an den NCR übertragen.
	wtk=default.wtk	Name und Pfadangabe einer Datei mit Werkzeugkorrekturdaten. Diese Datei wird nach einem vollständigen(!) Download zur Voreinstellung an den NCR übertragen.
	magazinplaetze=9	Angabe über die zur Magazinverwaltung freigestellten S-Koordinatensysteme (es stehen im NC maximal 100 S-Systeme gleichzeitig zur Verfügung).
	KSxx=a,c-e,g KSyy=e	Angabe über die Beziehung zwischen Magazinplatz (a, c-e,..) und korrespondierendem S-Koordinatensystem (xx und yy). Die Angabe kann in Form einer Aufzählung (a, c, d, e..) oder in Form einer Bereichsangabe erfolgen (c-e).
[SPS-Tasten]		In dieser Sektion können frei konfigurierbare ”SPS-Tasten”-Befehle eingetragen werden, die dann innerhalb der MMI-Software aus jedem Softkey Menü heraus durch einen gleichfalls konfigurierbaren ”Hotkey” aufgerufen werden können. Die Tasteninformationen (max. 128) werden der SPS über DB1 DW212-219 mitgeteilt.
	KeyCode=Shift-F8	Legt die Tastenkombination fest, auf die das alternative Softkeymenü mit den nachfolgend definierten SPS-Tasten-Funktionen auf dem Bildschirm erscheint bzw. wieder verschwindet. Eine gültige Angabe besteht aus der Festlegung einer Funktionstaste (F1 .. F12) und optional deren Verknüpfung mit einer beliebigen Kombination der Steuertasten ”Shift”, ”Alt” und ”Strg”.

Sektion	Term	Beschreibung
	KEYxxx=yyyy . . KEYmmm=nnnn,oooo	Festlegung der Eigenschaften der zu betätigenden ”SPS-Taste”: ● Verhalten als Taste bzw. Schalter, ● die Reihenfolge der Darbietung der Funktionen auf den Softkeys ● die Beschriftung der Softkeys werden über Einträge der Art KEYxxx= yyyy bzw. KEYmmm=nnnn,oooo erreicht. Der Wert ”xxx” beschreibt den Index eines Bits innerhalb der ”virtuellen Tastatur” (**DB1 DW212-DW219**) der SPS; der Wert ”yyyy”, ”nnnn” bzw. ”oooo” referiert auf Einträge in der Sprachdatei zur Beschriftung der entsprechenden Softkeys. Sind 2 mit Komma getrennte Sprachdateiverweise eingetragen (”nnnn,oooo”), verhält sich der Softkey wie ein Schalter. Es findet dann je nach Schaltzustand eine Umbeschriftung statt: ”nnnn” steht für den Schaltzustand ”aus”, ”oooo” für ”ein”. Die Reihenfolge der Einträge entspricht der der Darbietung der SPS-Tasten-Funktionen auf der horizontalen Softkeyzeile. Sind mehr als 8 Funktionen definiert, wobei ein Eintrag der Art KEYxxx=-1 eine leere Taste definiert, wird auf dem 8ten Softkey eine Funktion zum Fortschalten angeboten, z. B.: In der Ini-Datei steht: Key0 =2800,2927 Key12 =2818,2911 Key13 =-1 Key3 =2812 Key19 =2814 Key29 =2809 Key123 =2801 Key14 =2817 Key22 =2803 Key23 =2805 In der Sprachdatei steht: 00002800,” [0]@C12632256@FArial,8,1,0,3,6” 00002801,” [1]@C12632256@FArial,8,1,0,3,6” : 00002926,”[126]@C12632256@FArial,8,1,0,3,6” 00002927,”[127]@C12632256@FArial,8,1,0,3,6” Daraus ergeben sich 2 Softkeyzeilen mit folgendem Aufbau: [00] | [18] | | [12] | [14] | [09] | [01] |mehr.. [17] | [03] | [05] | | | | |mehr.. Nach Betätigung von ”[00]” wechselt die Beschriftung für Softkey 1 auf [127] [127] | [18] | | [12] | [14] | [09] | [01] |mehr. Beachten Sie bitte, dass der Zusammenhang zwischen der Softkey-Beschriftung, dem gesetzten Bit in der SPS-Tastatur und letztendlich der Reaktion an der Maschine nur durch die programmierten Abläufe in der SPS hergestellt wird.

Hinweis!

Man kann mit Hilfe der MMI-Software (Betriebsart Diagnose - Funktion MMI-Konfig.) die Datei bearbeiten. Zu beachten ist dabei, dass manche Änderungen erst nach einem Neustart der Software und manche Änderungen erst in Verbindung mit einem Zurücksetzen der Steuerung aktualisiert werden.

Einen Neustart der Software kann man erreichen, in dem zunächst die MMI-Software beendet wird und anschließend neu gestartet. Um die Steuerung zurückzusetzen, muss man sie aus- und einschalten oder den Reset-Taster betätigen.

SPS_EXEC

In der Delphmmi.ini im Abschnitt [SPS_EXEC] werden die Programme eingetragen, die ggf. ausgeführt werden sollen. Die entry-Nummer gibt den Index des Programms an, den die SPS mitteilen muss. Es wird der Name der Applikation eingetragen. Mit Semikolon getrennt kann dann noch ein Programm-Parameter angegeben werden.

Dahinter kann noch angegeben werden, ob die Applikation in einem normalen Fenster (NORMAL), maximiert oder minimiert ausgeführt werden soll. Standard ist normal.

```
[SPS_EXEC]
entry0=C:\WinNT\Notepad
entry1=C:\WinNT\Notepad;Readme.txt
entry2=C:\WinNT\Notepad;Readme.txt;Max
entry3=C:\WinNT\Notepad;;Min
```

Die SPS sendet nun eine Nachricht **SB0_PLC_EXECUTE = $8015;** an das MMI. Im ersten Wort der Nachricht steht die Entry-Nummer. Ist das zweite Datenwort ungleich Null, wird bei einem Fehler ein Dialogfenster angezeigt.

Das HMI sendet der SPS immer eine Nachricht zurück: **SB0_MMI_EXECUTE = $8016;**. Im ersten DW ist die Fehlernummer codiert.

0: kein Fehler
1: Fehler bei Ausführung des Programms
2: Es läuft bereits ein Programm
3: Programm nicht gefunden
4: Kein Programm für den Entry eingetragen

Im zweiten DW wird die Nummer des angeforderten Jobs zurückgegeben.

Hinweis!

Einschränkung: Es kann immer nur ein Programm aus der Liste ausgeführt werden.

BARANZ

Die Balkenanzeigen für Vist und Override können umkonfiguriert werden, um einen anderen Wert aus dem DPR angezeigt zu bekommen. Des Weiteren können in der Betriebsart Automatik zwei zusätzliche Balkenanzeigen unterhalb der Achspositionen konfiguriert werden.

Dazu gibt es einen neuen Abschnitt [BarAnz] in der Delphmmi.ini. Dieser Abschnitt darf bis zu vier Einträge für die vier Balken enthalten.

Beispiel:

```
[BARANZ]
ANZ1=0;800;10802;1;P30
ANZ2=0;800;10902;2;P31
ANZ3=0;3000;10920;3;4711;DOUBLE
Anz4=0;4000;10930;4;D10,WORD
```

Die Werte eines Eintrags haben folgende Bedeutung:

1. Min-Wert des Balkens bzw. der Beschriftung (derzeit werden Einträge < 0 nicht unterstützt)
2. Max-Wert des Balkens bzw. der Beschriftung (der Max-Wert muss größer als der Min-Wert sein)
3. Textnummer des beschreibenden Textes in der Sprache Text
4. Balken auf dem der Wert angezeigt werden soll:
   1= rechts oben (Standard Vist)
   2= rechts unten (Standard Override)
   3= links oben (Standard unsichtbar)
   4= links unten (Standard unsichtbar)
5. Anzuzeigender Wert; es gibt drei Arten einen Wert anzugeben:
   PXX: Index auf die Feld-Permanentanzeige; das entsprechende Feld muss in der Delphmmi.ini in für die entsprechenden Anzeigen eingetragen werden.
   DXX: SPS DB2 Anzeige aus 0 ... 63,
   XXXX: Direkt die Adresse im DPR
6. Wenn die Angabe des anzuzeigenden Wertes über DB2 bzw. über eine direkte Adresse angegeben wird, muss noch das Wertformat angegeben werden. Folgende Werte sind erlaubt:
   BYTE
   DOUBLE
   INTEGER
   LONG
   SINGLE
   WORD

Bei P-Feldern ist der Datentyp immer double.

8 SPS-Programmierung

8.1 ETC-SPS-Programmierung mit CoDeSys

CoDeSys ist eine komplette Entwicklungsumgebung zum Erstellen und Testen von SPS-Programmen für die ETC.

CoDeSys bietet Möglichkeiten zum Debuggen von Programmen, ähnlich den heutigen Hochsprachenentwicklungssystemen (Setzen von Haltepunkten, Monitoring von Variablen, Aufzeichnung eines Traces (Oszilloskopfunktion) usw.).

Eine Einführung in die SPS-Programmierung nach IEC 61131-3 mit CoDeSys enthält das CoDeSys-Handbuch (siehe ”Docu”-Verzeichnis im CoDeSys-Installationspfad).

Dieses Benutzerhandbuch ist als Ergänzung zum CoDeSys-Handbuch zu verstehen. Es enthält Beschreibungen zu Funktionen bzw. Anpassungen im Zusammenhang mit CoDeSys und der ETC.

8.2 CoDeSys installieren

8.2.1 Systemvoraussetzungen für CoDeSys V2.xx

- Pentium-Prozessor (Pentium II, ab 350 MHz empfohlen)
- 32 MB RAM (64 MB empfohlen)
- Windows 2000 oder XP
- MS Internet Explorer ab Version 4.0

8.2.2 Software installieren

1. Legen Sie die CoDeSys-Setup-CD in Ihr CD-ROM-Laufwerk.
   Wenn das Autostart-Feature des CD-ROM-Laufwerks in Windows aktiviert ist, startet das Setup automatisch.
2. Sollte das Setup nicht automatisch starten:
   – Wählen Sie im Startmenü den Menüpunkt ”Ausführen”.
   – Geben Sie in der Befehlszeile den Laufwerksbuchstaben Ihres CD-ROM-Laufwerks gefolgt von ”setup.exe” ein (z. B. ”d:\setup.exe”) und bestätigen Sie mit ”OK”.
3. Bei der Auswahl des Zielverzeichnisses generieren Sie bitte ein neues Verzeichnis ”CoDeSys” unter ”c:\programme\Lenze\ETC”.
4. Folgen Sie den Anweisungen des Setup-Programms.
5. Ändern Sie den Installationspfad in ”c:\programme\lenze\etc\CoDeSys2”.

Stop!

Die Datei ”codesys.ini” im Installationsverzeichnis kann nach der Installation von CD schreibgeschützt sein. Der Schreibschutz muss aufgehoben werden, da sonst die CoDeSys-Einstellungen nicht gespeichert werden.

8.3 ETC und PC verbinden

Die Verbindung zur ETC kann über eine der folgenden Schnittstellen hergestellt werden:

- ► Serielle Schnittstelle
- ► Ethernet-Schnittstelle (nur ETCHx, Hutschienen-Ausführung)
- ► DPR-Schnittstelle (nur ETCPx, PCI-Steckkarte)

8.3.1 V.24-Schnittstelle

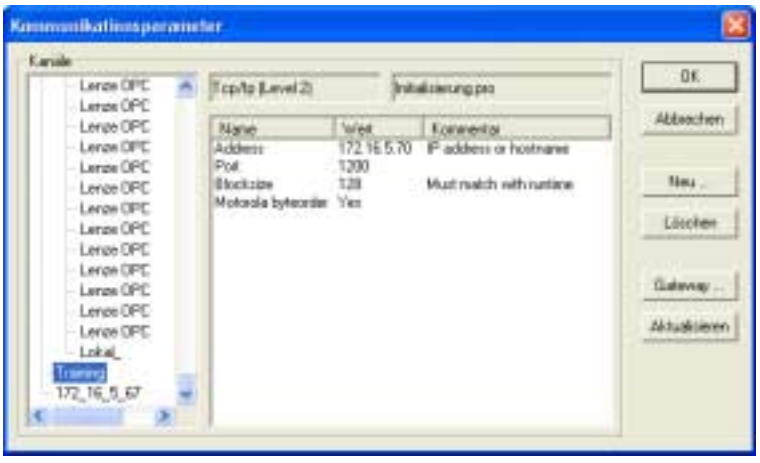
Für die Kommunikation über die serielle Schnittstelle muss der RS232- oder CAN2-Anschluss der ETC über ein Nullmodemkabel mit einem freien COM-Port des PCs verbunden werden. In den Kommunikationsparametern von CoDeSys muss der Treiber ”3S Serial RS232 driver” ausgewählt werden. In den Einstellungen zu diesem Treiber müssen folgende Werte für die entsprechende Schnittstelle der Steuerung ausgewählt werden:

Anschluss ETC	RS232	CAN2
Baudrate	115200	38400
Parity	No	No
Stop bits	1	1
Motorola byteorder	Yes	Yes

8.3.2 Ethernet-Schnittstelle (nur ETCHx, Hutschienenausführung)

Für die Kommunikation über Ethernet muss der Ethernet-Anschluss am ETCHx über ein Ethernetkabel mit dem PC verbunden werden.

In den Kommunikationsparametern von CoDeSys muss der Treiber ”3S Tcp/Ip level2 driver” ausgewählt werden. In den Einstellungen zum Treiber müssen folgende Werte eingetragen werden:

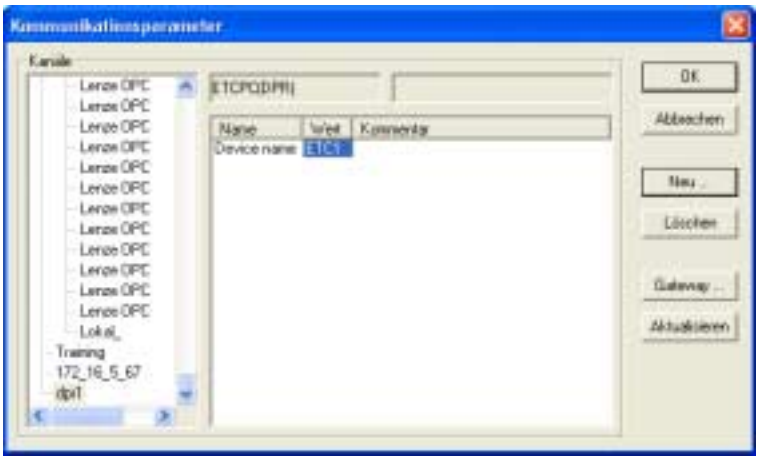


ETC100

8.3.3 DPR-Schnittstelle (nur ETCPx, PCI-Steckkarte)

Für die Kommunikation über die DPR-Schnittstelle ist es notwendig, dass der Windows WDM-Treiber für die ETCPx und der Lenze MMI Gateway installiert sind.

In den Kommunikationsparametern von CoDeSys muss der Treiber ”E*DPR” ausgewählt werden. In den Einstellungen zu dem Treiber müssen folgende Werte eingetragen werden:



ETC101

Bei Auswahl des Treibers ”DPR” können die im MMI-Gateway angegebenen Steuerungen unter dem Tabelleneintrag ”Wert” ausgewählt werden.

Dies geschieht durch Doppelklicken auf das entsprechende Feld (hier: ETC1) und anschließendem Umschalten über die Pfeiltasten.

8.4 Projektierung

8.4.1 Zielsystem einstellen

Beim Anlegen eines neuen Projekts in CoDeSys öffnet sich automatisch ein Dialog, der nach der verwendeten Hardware (=Zielsystem) fragt. Nach dem Anlegen ist der Konfigurationsdialog über den Menüpunkt ”Zielsystemeinstellungen” im Registerblatt ”Ressourcen” aufrufbar.

Hinweis!

Das gewählte Zielsystem wird beim Einloggen in die Steuerung überprüft, so dass ein SPS-Programm durch CoDeSys nicht zu einem falschen Zielsystem übertragen werden kann.

Bei einem Bootprojekt (📖 349) ist darauf zu achten, dass das ausgewählte Zielsystem zu der verwendeten Firmware der Steuerung passt, da beim Laden eines Bootprojekts, zum Beispiel durch ein ETC-MMI, dieses nicht überprüft wird.

► Steuerungsvariante CNC: Zielsystemvorwahl = ETCxC

► Steuerungsvariante Motion: Zielsystemvorwahl = ETCxM

Die Parameter unter ”Zielsystemeinstellungen” werden anhand des gewählten Zielsystems voreingestellt und brauchen in der Regel nicht verändert werden.

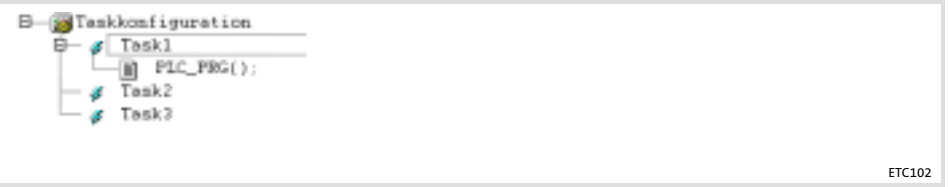
Eine Beschreibung der Zielsystemeinstellungen finden Sie im CoDeSys-Handbuch.

8.4.2 SPS-Tasks der ETCxM konfigurieren

SPS-Tasks einbinden

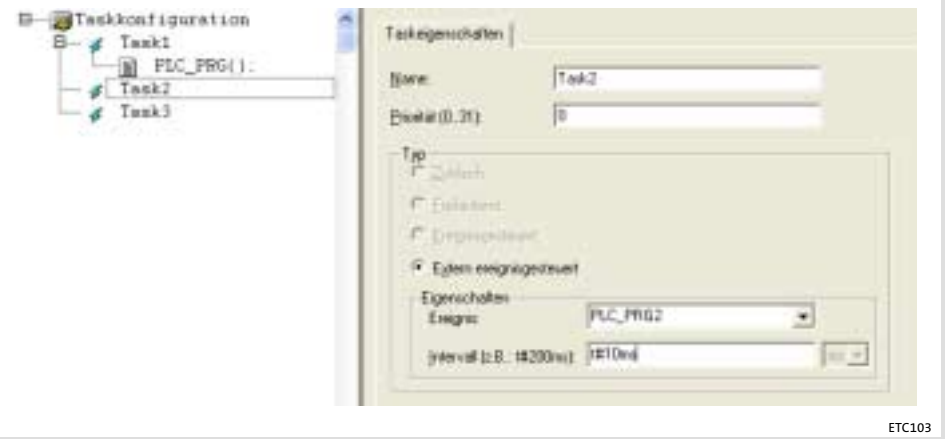
Mit der ETCxM ist es möglich, vier voneinander unabhängige SPS-Tasks zu nutzen. Jede Task kann mit einem Aufruf eines IEC-Programmbausteins (PRG) verknüpft sein. Die mit den Tasks verknüpften Programmbausteine werden von dem Multitaskingbetriebssystem der ETCxM quasiparallel bearbeitet.

Die Abarbeitungsreihenfolge ergibt sich aus der Intervallzeit (zeitgesteuert) und der Priorität einer Task. Eine Task läuft also solange, bis ein SPS-Zyklus abgearbeitet wurde oder eine andere Task mit höherer Priorität gestartet wird. Damit ein SPS-Programm durch die ETCxM abgearbeitet wird, muss in der Taskkonfiguration von CoDeSys mindestens eine Task angelegt sein und mit einem IEC-Programmbaustein verknüft werden.



Eigenschaften einer Task

Die Eigenschaften der einzelnen Task konfigurieren Sie in der ”Taskkonfiguration” von CoDeSys.



Folgende Punkte sind dabei zu beachten:

- ► Je größer der Wert bei Priorität, desto größer ist die Task-Priorität.
- ► Eine Task mit Intervall von ”t#0s” darf höchstens eine Priorität von 10 erhalten, sonst setzt die Steuerung die Priorität aller Tasks auf 0 und meldet einen Fehler (Modul 13, Fehlernummer 202).
- ► Zudem muss eine Task mit Intervall von ”t#0s” die niedrigste Priorität von allen Tasks haben, sonst setzt die Steuerung die Priorität aller Tasks auf 0 und meldet einen Fehler (Modul 13, Fehlernummer 201). Die Tasks mit niedrigerer Priorität als die vorgenannte Tasks würden sonst nicht zur Ausführung kommen.
- ► Gibt es mehrere Tasks mit Intervall ”t#0s”, müssen diese Tasks alle die gleiche Priorität haben.
- ► Die Intervallzeit ist zwischen ”t#0ms” (Dauerläufer) und ”t#1000ms” zu wählen.
- ► Bei dem Eintrag ”Ereignis” in den Taskeigenschaften von CoDeSys ist jede Task mit einer Ereignisvariablen zu verknüpfen.
- ► Das Betriebssystem der ETCxM ordnet damit den 4 Tasks den entsprechenden Aufruf des mit der Task verknüpften CoDeSys-Programmbausteins zu. Jeder Task ist eine Ereignisvariable zugeordnet.

Tasknummer	Ereignisvariable
1	PLC_PRG1
2	PLC_PRG2
3	PLC_PRG3
4	PLC_PRG4

Stop!

Beim Zugriff auf Daten bzw. beim Aufruf von Bausteinen, die in mehreren Tasks verwendet werden, muss beachtet werden, dass keine Synchronisation zwischen den Tasks besteht.

Der SPS-Programmierer muss dafür Sorge tragen, dass sich dadurch keine Probleme ergeben

Prozessabbild

Bei der Zuordnung der CoDeSys-Programmbausteine zu den SPS-Tasks ist zu beachten, dass jeder Task nur ein Teil des Prozessabbildes zugeordnet ist. Das Prozessabbild ist eine Kopie der aktuellen Zustände der Ein-/Ausgangsbaugruppen (Hardware), auf die das SPS-Programm zugreift.

Vor jedem SPS-Zyklus (kompletter Programmdurchlauf) werden die aktuellen Zustände der Eingangsmodule in das Prozessabbild kopiert und nach jedem SPS-Zyklus werden die aktuellen Zustände der Ausgänge im Prozessabbild zu den Ausgangsmodulen übertragen.

Da jeder Task ein Teil des Prozessabbildes zugeordnet ist, wird vor und nach einem Durchlauf des Programmbausteins, der einer Task zugeordnet ist, nur der entsprechende Teil des Prozessabbildes aktualisiert.

8.4.3 SPS-Tasks der ETCxC konfigurieren

SPS-Tasks einbinden

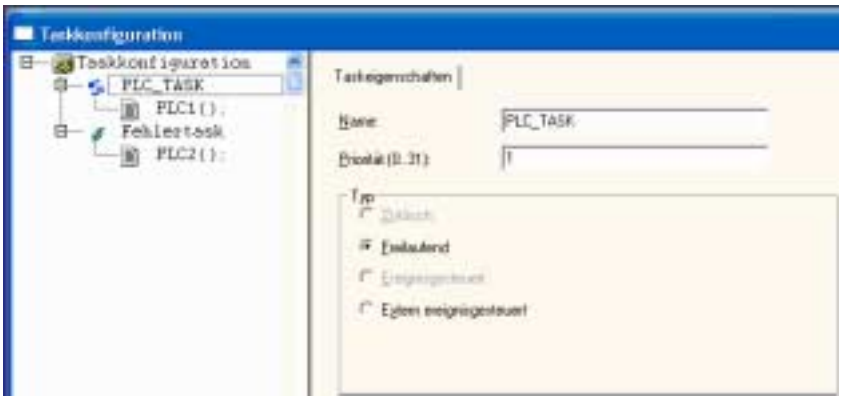
Die ETCxC besitzt im Gegensatz zur ETCxM nur 2 Tasks, wobei Task 1 die Haupttask ist. Die 2. (niederpriore) Task dient als Schnittstelle zu steuerungsinternen Signalen zur Bedienung (z. B. Fehlermanagement).

Um einen benutzerspezifischen Baustein (IEC-Programmbaustein: PROGRAM) innerhalb der Task anstatt der steuerungsinternen Funktion ablaufen zu lassen, sind folgende Schritte notwendig:

Schritt 1:

In der Taskverwaltung von CoDeSys sind zwei Tasks einzufügen:

1. Taskeintrag:



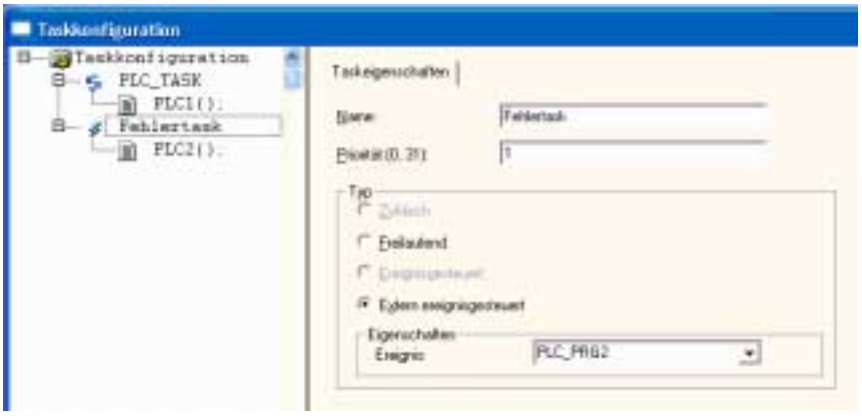
ETC104

Schritt 2:

Zur Kennzeichnung für die Steuerung, dass es sich bei diesem Taskeintrag um die 2. (niederpriore) Task handelt, ist im Feld ”Ereignis” ”PLC_PRG2” einzugeben (oder OB20, wenn der Programmbaustein, der mit der Task verknüpft werden soll ”PLC_PRG2” benannt wurde).

2. Taskeintrag:

Sobald die Taskverwaltung von CoDeSys genutzt wird, muss auch ein Taskeintrag für den Hauptbaustein ”PLC_PRG” eingetragen werden, da sonst dieser Baustein nicht mehr aufgerufen wird:



ETC105

Wird nur die Haupttask benötigt, ist eine Editierung der Taskkonfiguration nicht notwendig.

Schritt 3:

Verknüpfen Sie die Tasks mit IEC-Programmbausteinen.

Mit dem 1. Taskeintrag kann ein beliebiger IEC-Programmbaustein verknüpft werden. Da der Bausteinbezeichner ”PLC_PRG” eine besondere Bedeutung hat, sollte dieser umbenannt werden (z. B. ”PLC_PRG1”) und dann mit dem 2. Taskeintrag verknüpft werden.

Hinweis!

Beim Zugriff auf Daten bzw. beim Aufruf von Bausteinen, die in beiden Tasks (PLC_TASK (PLC_PRG1), z. B. Name: BDT_TASK (PLC_PRG2)) verwendet werden, muss beachtet werden, dass keine Synchronisation zwischen beiden Tasks besteht.

Der SPS-Programmierer muss dafür Sorge tragen, dass sich dadurch keine Probleme ergeben.

8.4.4 E/A-Module konfigurieren

CoDeSys beinhaltet einen CAN-Bus-Konfigurator. Der Konfigurator stützt sich auf sogenannte EDS-Dateien (EDS = Electronic Data Sheet), die im Unterverzeichnis ”ETC” des Targets-Verzeichnisses abgelegt sein müssen.

Eine allgemeine Beschreibung zur Konfiguration eines CAN-Moduls finden Sie in der CoDeSys-Dokumentation.

8.4.4.1 CAN-Master konfigurieren (globale CAN-Einstellungen)

Basisparameter

Adresse automatisch: Der Schalter sollte abgewählt werden, da CoDeSys sonst die Adressen automatisch vergibt.

CAN-Parameter

Baudrate: Die Baudrate für den CAN-Bus muss entsprechend den Einstellungen der angeschlossenen CAN-Module ausgewählt werden.

Hinweis!

Die Eingabe der Baudrate ist nur für die ETCxM aktiv.

Für die ETCxC hat dieser Parameter keine Funktion. Es gilt nur die Einstellung in den Maschinenkonstanten.

Com. Cycle Period: Wird von der Steuerung nicht ausgewertet.

Sync. Window Length: Wird von der Steuerung nicht ausgewertet.

Sync COB-ID: Wird von der Steuerung nicht ausgewertet.

Node-ID: Geben Sie hier eine ”0” ein.

Automatisch Starten: Wird von der Steuerung nicht ausgewertet.

DSP301 ...: Wird von der Steuerung nicht ausgewertet.

Heartbeat Master: Wird von der Steuerung nicht ausgewertet.

8.4.4.2 CAN-Slave konfigurieren

Basisparameter

Modul-ID: Idennummer des Moduls.

Knotennummer: Die Node-ID wird unter der Registerkarte CAN-Parameter eingegeben.

Eingabe-/Ausgabeadresse ETCxM: Die Basisadressen sind immer als Wortadressen anzugeben. Mit der Adresse wird festgelegt, an welcher Stelle die Prozessdaten des Moduls im Prozessabbild (E/A-Speicher) liegen sollen. Damit ist auch verbunden, welcher SPS-Task die Daten zugeordnet werden. Die folgende Tabelle zeigt die Zuordnung des Prozessabbildes zu den einzelnen SPS-Tasks :

Tasknr.	E/A-Bereich	
	von	bis
1	%W0	%W1023
2	%W1024	%W1535
3	%W1536	%W1791
4	%W1792	%W2047

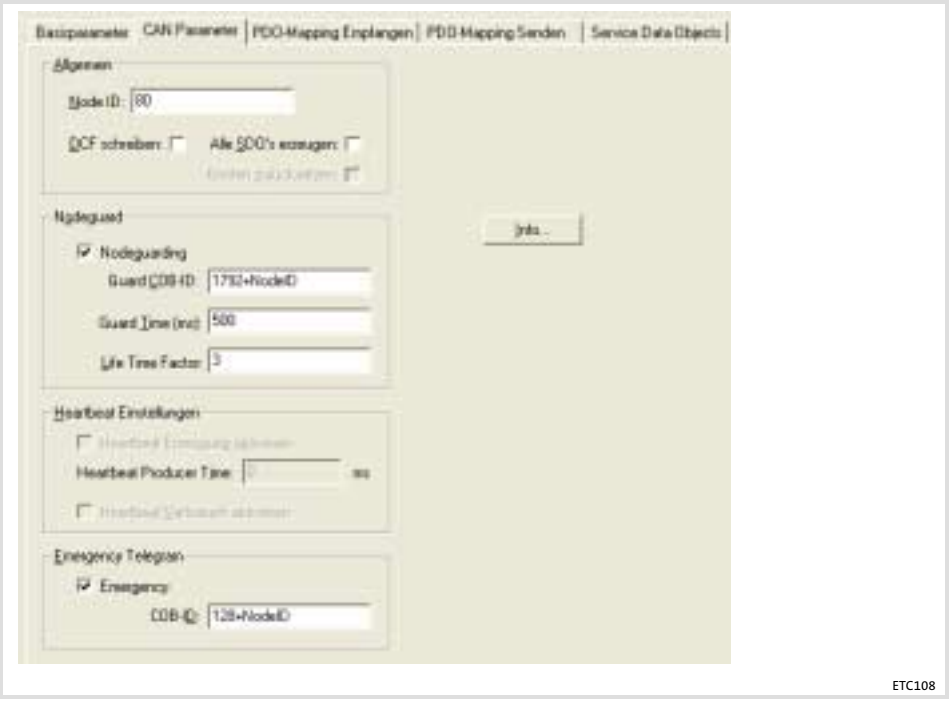
Weitere Informationen finden Sie im Kapitel ”Adressierung” (📖 342).

Eingabe-/Ausgabeadresse ETCxC: Die Basisadressen sind als Wortadressen im Bereich von %QW64 / %IW64 bis %QW127 / %IW127 zu vergeben.

Weitere Informationen finden Sie im Kapitel ”Adressierung” (📖 342).

Diagnoseadresse: Dieses Feld wird von der Steuerung nicht ausgewertet.

CAN-Parameter



Allgemein: Bei Node-ID ist die an dem Modul eingestellte CAN-ID (Knotennummer) einzutragen.

Nodeguarding: Das Nodeguarding dient zum Erkennen, ob ein CANopen-Modul am Bus angeschlossen ist. Dazu sendet die Steuerung im Intervall von ”Guard time” eine Nachricht zum Modul und erwartet eine Antwort.

Empfängt die Steuerung nach ”Life-Time Factor”-Anforderungen keine Antwort, wird eine Fehlermeldung erzeugt.

Ist keine ”Guard Time” und kein ”Life Time Factor” angegeben, wird das CAN-Modul nicht überwacht; das heißt, ein Ausfall wird nicht bemerkt.

Heartbeat Einstellungen: Wird von der Steuerung nicht ausgewertet.

Emergency: Ist dieser Eintrag abgewählt, ignoriert die Steuerung ”Emergency-Telegramme” der Steuerung.

PDO-Mapping

Per PDO (Process Data Object) werden bei CANopen die Prozessdaten übertragen; das heißt die Zustände der digitalen und analogen Ein-/Ausgänge.

Jeder PDO ist eine eindeutige COB-ID (385-1407) zugewiesen, wobei die COB-IDs bei den CANopen-IO-Modulen mit der Annahme von maximal 127 Modulen folgendermaßen vergeben sind:

1. Tx-PDO: 384 + NodeID
1. Rx-PDO: 512 + NodeID
2. Tx-PDO: 640 + NodeID
2. Rx-PDO: 768 + NodeID
3. Tx-PDO: 896 + NodeID
3. Rx-PDO: 1024 + NodeID
4. Tx-PDO: 1152 + NodeID
4. Rx-PDO: 1280 + NodeID

(Tx = Transmit, Rx = Receive ist aus Sicht des CANopen-Moduls zu sehen)

Es können also ohne Einschränkung der maximalen Anzahl von Modulen 4 Receive- und Transmit-PDOs pro Modul vergeben werden. Sollen abweichend vom Standard mehr PDOs pro Modul definiert werden, ist darauf zu achten, dass die COB-ID's nicht doppelt vergeben werden.

Die erste Tx/Rx-PDO enthält die digitalen Daten und die zweite Tx/Rx-PDO die analogen Daten. Für die zweite PDO sollte bei "Transmission Type" immer "asynchron-nur RTR" eingetragen sein, da dies die Standardeinstellung der meisten Module für die Übertragung von analogen Werten ist.

Service Data Objects

Diese Einträge werden von der Steuerung nicht ausgewertet.

8.4.4.3 Überprüfung der Anzahl von Eingängen und Ausgängen

Die Steuerung führt eine implizite Überprüfung der Anzahl der digitalen/analogen Ein- und Ausgänge durch.

Dabei wird von der Annahme ausgegangen, dass die digitalen Daten in der ersten Tx/Rx-PDO und die analogen Werte in der zweiten Tx/Rx-PDO liegen. Diese Annahme und die daraus folgende Überprüfung wird nur durchgeführt, wenn in der EDS-Datei zu dem CANopen-Modul für den Objekteintrag 1000_{hex} (Device Profil Number) in den zusätzlichen Informationen (höchstwertigen 16-bit) die IO-Unterstützung für digitale und analoge Ein-/Ausgänge markiert ist (s. CiA DS401: Device Profil for Generic I/O Modules).

Hinweis!

Die Steuerung erzeugt eine Fehlermeldung, wenn die Anzahl der konfigurierten digitalen/analogen Ein-/Ausgänge einer PDO die tatsächliche Anzahl übersteigt.

Auszug aus einer EDS-Datei

```
[1000]
ParameterName=DeviceType
ObjectType=0x7
DataType=0x7
AccessType=ro
PDOMapping=0
DefaultValue=0x30191
```

Beschreibung Objektindex 1000hex

Zusätzliche Information				Allgemeine Information	
Spez. Funktionalität		I/O Funktionalität		Device Profile Number	
31	24	23	16	15	0

I/O-Funktionalität

Bit 16: Digitale Eingänge
Bit 17: Digitale Ausgänge
Bit 18: Analoge Eingänge
Bit 19: Analoge Ausgänge

Überprüfung ausschalten

Um die Überprüfung der Anzahl der IOs auszuschalten, muss DefaultValue = 0x191 gesetzt werden. Falls ein Modul auch mit der 1. Tx/Rx-PDO analoge Daten überträgt, muss DefaultValue = 0xC0191 gesetzt werden, also die Unterstützung für digitale E/A's ausgeschaltet werden.

8.4.4.4 Suchen von konfigurierten CANopen-Modulen

Die Steuerung sucht nach allen in der Steuerungskonfiguration konfigurierten Modulen bevor die eigentliche Kommunikation (per PDO) mit dem Modul gestartet wird.

Das Suchen der Module erfolgt, per NodeGuarding, falls konfiguriert, ansonsten per SDO-Transfer (Auslesen von Objektindex 1000_{hex}).

Suchen ausschalten

Für CANopen-Module, die kein NodeGuarding oder SDO-Transfer unterstützen, kann das Suchen der Module ausgeschaltet werden. Dazu muss in der EDS-Datei zu dem CANopen-Modul für den Objekteintrag 1000_{hex} (Device Profil Number) eine 0 eingetragen werden.

8.4.5 Adressierung

Die ETC arbeitet nach dem Big-Endian-Datenmodell, dabei liegen alle Datentypen mit dem hochwertigsten Byte beginnend, also dem ”dicken Ende”, zuerst im Speicher. Dies muss vor allem bei der Kommunikation mit einem PC-HMI (z. B. ETC-MMI) über den DB2 (ETCxC) berücksichtigt werden, da PC’s nach dem Little-Endian-Datenmodell arbeiten. Die Steuerung stellt hierfür einige Funktionen zur Verfügung, um dies zu erleichtern (📖 384).

Die direkte Darstellung einer einzelnen Speicherzelle erfolgt mittels spezieller Zeichenreihen. Diese entsteht aus dem Prozentzeichen ”%”, einem Bereichspräfix, einem Präfix für die Datenbreite und zwei oder drei natürlichen Zahlen, die durch Punkte voneinander getrennt sind.

Bereichspräfix	Bedeutung
I	Eingang
Q	Ausgang
M	Datenbaustein (DBO ... DB15) der ETC

Datenbreiten-Präfix	Bedeutung
X	1 Bit
B	Byte (8 Bits)
W	Word (16 Bits)
D	Doppelwort (32 Bits)

8.4.5.1 E/A-Module adressieren

Die E/A-Adressierung ist grundsätzlich wortweise organisiert. Die Adressen der I/O’s ergeben sich aus der internen Organisation des Prozessabbildes der Steuerung.

Zugriffsart	Syntax	Bemerkung
Bitweise	%IX x.y / %QX x.y	x: Nummer des Datenwortes (0 ... 127) y: Bit in dem Wort (0 ... 15)
Byteweise	%IB x / %QB x	x: Nummer des Bytes (0 ... 254)
Wortweise	%IW x / %QW x	x: Nummer des Datenwortes
Doppelwortweise	%ID x / %QD x	x: Nummer des Doppelwortes

Stop!

Wortweiser Zugriff: Wegen der internen Datenorganisation ist bei einem byteweisen Zugriff darauf zu achten, dass im Speicher zuerst das High-Byte und dann das Low-Byte steht (siehe Beispiel).

Doppeltwortweiser Zugriff: Wegen der internen Datenorganisation macht ein doppelwortweiser Zugriff normalerweise keinen Sinn, da High- und Low-Word vertauscht wären (siehe Beispiel).

8.4.5.2 Datenbausteine adressieren

Die Datenbausteine (DB0 - DB15) werden über den Bereichspräfix "M" angesprochen.

Zugriffsart	Syntax	Bemerkung
Bitweise	%MX x.y.z	x: Nummer des Datenbausteins y: Wort in dem Datenbaustein (0 ... 255) z: Bit in dem Wort (0 ... 15)
Byteweise	%MB x.y.z	x: Nummer des Datenbausteins y: Wort in dem Datenbaustein (0 ... 255) z: linkes (höherwertiges) oder rechtes (niederwertiges) Byte des Wortes (links = 1, rechts = 0)
Wortweise	%MW x.y	x: Nummer des Datenbausteins y: Wort in dem Datenbaustein (0 ... 255)
Doppelwortweise	%MD x.y	x: Nummer des Datenbausteins y: Doppelwort in dem Datenbaustein (0 ... 127)

Stop!

Doppeltwortweiser Zugriff: Wegen der internen Datenorganisation macht ein doppelwortweiser Zugriff normalerweise keinen Sinn, da High- und Low-Word vertauscht wären (siehe Beispiel).

8.4.6 Remanente Variablen

Mit dem Schlüsselwort "RETAIN" können bei der Variablendeklaration in CoDeSys remanente Daten festgelegt werden (siehe CoDeSys-Dokumentation). Es stehen insgesamt 12 kB für remanente Daten zur Verfügung.

8.4.7 Objektverzeichnis (Parametermanager)

In CoDeSys ist es möglich, ein sogenanntes CANopen-Objektverzeichnis anzulegen, das durch ein CANopen-Gerät (z. B. CANopen-Bedienteil) gelesen und beschrieben werden kann.

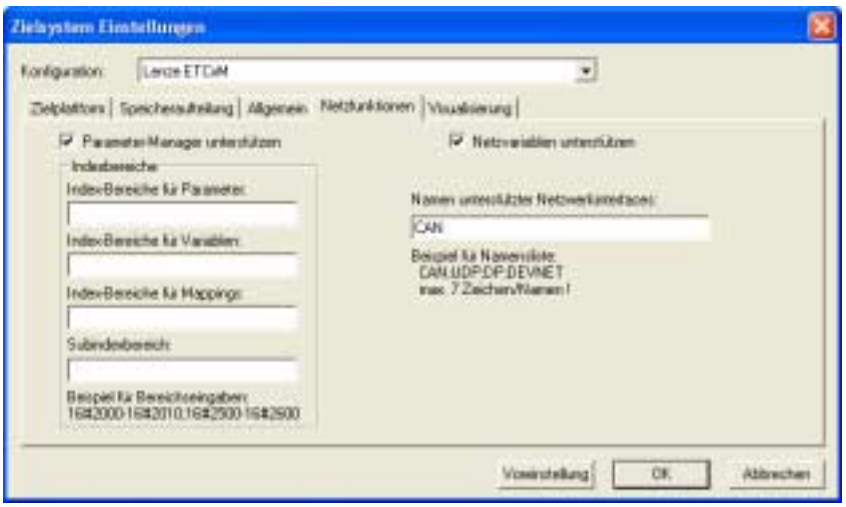
Um die Übertragung eines Objektverzeichnisses über den CAN-Bus zu aktivieren, muss der Steuerung über die Funktion "InitServerSdo()" (siehe Bibliothek "ServerSdo.lib") das Objektverzeichnis bekannt gegeben werden.

Hinweis!

Beim Einfügen der Bibliothek "ServerSdo.lib" in der Bibliotheksverwaltung von CoDeSys werden automatisch weitere Bibliotheken (SysExtensions.lib, SysLibCallback.lib, Netvarcan_Lib.lib) eingefügt.

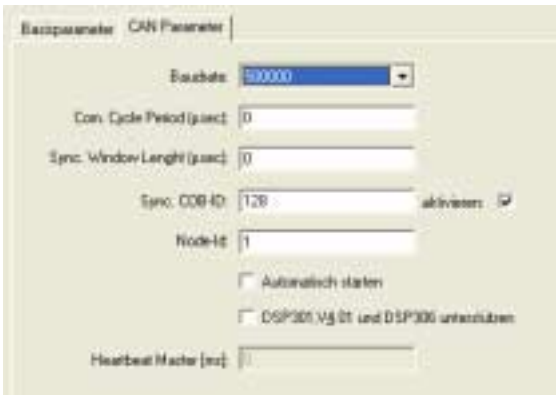
Ein Objektverzeichnis kann über den Objektverzeichnis-Editor von CoDeSys erstellt werden. Um den Objektverzeichnis-Editor benutzen zu können, muss im CoDeSys unter **ZielsystemEinstellungen -> Netzfunktionen** (siehe folgendes Bild) die Unterstützung von Objektverzeichnissen und Netzvariablen aktiviert sein.

Definieren Sie Index- bzw. Subindex-Bereiche für SDO-Parameter und Variablen und geben Sie ”CAN” für den Namen unterstützter Netzwerkinterfaces ein.



ETC109

Die CANopen-Adresse der Steuerung ist unter **Steuerungskonfiguration -> CAN-Master -> CAN Parameter->Node-Id** festzulegen.



ETC116

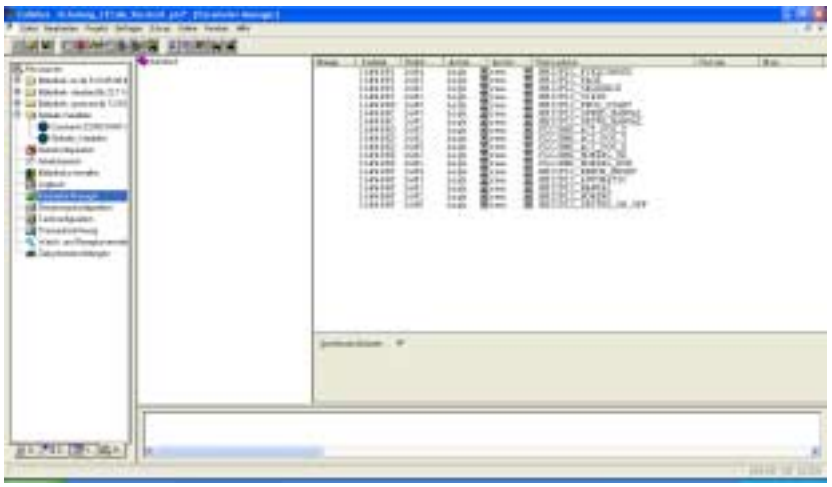
Für die Festlegung der Node-Id ist Folgendes zu beachten:

Wenn die Node-Id im Bereich 1 ... 63 liegt, werden zwei Server-SDO-Kanäle mit den Adressen ”Node-Id” und ”Node-Id + 64” erzeugt, d. h. die Steuerung erhält 2 CANopen-Adressen. Da die Verwendung von 2 SDO-Kanälen im Allgemeinen nicht notwendig ist, sollte eine Node-Id im Bereich 64 ... 127 vergeben werden, da dann nur ein SDO-Kanal angelegt wird.

Im Objektverzeichnis-Editor von CoDeSys dürfen nur Einträge in der Registerkarte ”Variable” eingetragen werden.

8.4.8 Einstellungen in den globalen Variablenlisten im Objektverzeichnis

Legen Sie ein neues Objektverzeichnis an. Hier definieren Sie die Variablen, die mit anderen Steuerungen ausgetauscht werden sollen.


ETC110

Die Registerkarte Variable muss angewählt sein. Nun können die Indexe mit ihren Variablennamen eingeben werden.

Index: In hexadezimaler Schreibweise

Subindex: Der Bereich der Subindexe ist in den Zieleinstellungen festgelegt.

Zugriff: Zugriffshäufigkeit

Attribut: Eigenschaft der Variablen (Schreiben, Lesen)

Variable: Name der Variablen (Achtung: vor dem Text muss ein Punkt stehen.)

8.5 Netzwerkvariablen

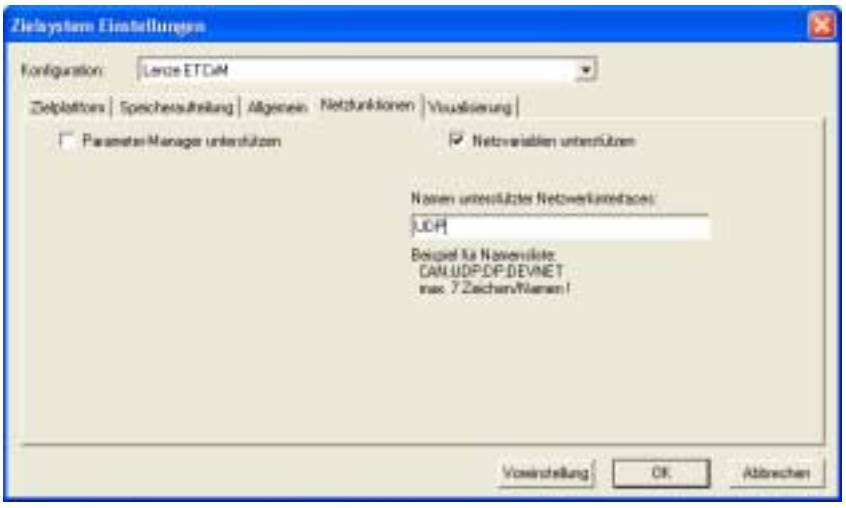
Netzwerkvariablen sind eine Möglichkeit, Daten zwischen zwei oder mehreren Steuerungen auszutauschen. Derzeit sind Netzwerkvariablen auf Basis von UDP implementiert. Die Variablenwerte werden dabei auf der Basis von Broadcast-Nachrichten automatisch ausgetauscht. Diese Dienste sind vom Protokoll her nichtbestätigte Dienste, d. h. es gibt keine Kontrolle, ob die Nachricht auch beim Empfänger ankommt. Der Netwerkvariablenaustausch entspricht einer 1 (Sender) zu n (Empfänger)-Verbindung.

Voraussetzung

Die Bibliotheken ”NetVarUdp_LIB.lib”, ”SysLibCallback.lib” und ”SysLibSocket.lib” sind in der Bibliotheksverwaltung von CoDeSys eingefügt.

8.5.1 Einstellungen in den Zielsystemeinstellungen

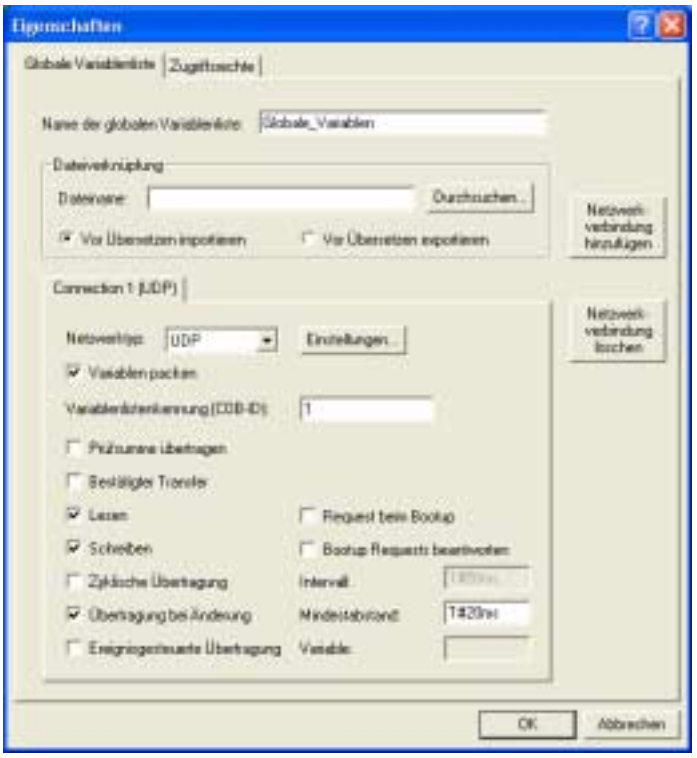
Aktivieren Sie das Kontrollkästchen ”Netzvariablen unterstützen” im Reiter ”Netzfunktionen” der Dialogbox ”Zielsystemeinstellungen”. Bei ”Namen unterstützter Netzwerkinterfaces” geben Sie den ”UDP” an.



ETC111

8.5.2 Einstellungen in den globalen Variablenlisten

Legen Sie eine neue globale Variablenliste an. Hier definieren Sie die Variablen, die mit anderen Steuerungen ausgetauscht werden sollen. Über den Eigenschaftsdialog der Variablenliste lassen sich die Übertragungseigenschaften festlegen.



Wenn Sie die Schaltfläche **Netzwerkverbindung hinzufügen** drücken, können Sie die Netzwerkeigenschaften dieser Variablenliste definieren. Die Optionen haben dabei folgende Bedeutung:

Netzwerktyp: Als **Netzwerktyp** muss UDP ausgewählt werden.

Lesen: Lesen bedeutet, dass die Variablenwerte von einer (oder mehreren) anderen Steuerungen gelesen werden.

Schreiben: Schreiben bedeutet, dass die Variablen dieser Variablenliste zu anderen Steuerungen gesendet werden. Es wird empfohlen, für eine Variablenliste nur eine der Möglichkeiten zu wählen, also entweder nur Lesen oder nur Schreiben. Sollen verschiedene Variablen eines Projekts gelesen und geschrieben werden, so verwenden Sie mehrere Variablenlisten: eine zum Lesen, eine zum Schreiben. Außerdem wird empfohlen, dass in einem Netzwerk nur eine Steuerung die gleiche Variablenliste sendet.

Zyklische Übertragung: Nur gültig, wenn ”Schreiben” aktiviert ist. Die Werte werden in der angegebenen Intervallzeit gesendet, unabhängig davon, ob sie sich geändert haben.

Übertragung bei Änderung: Die Werte werden nur gesendet, wenn sich der Wert geändert hat, oder wenn die bei Mindestintervall angegebene Zeit abgelaufen ist.

Variablen packen: Wenn diese Option aktiviert ist, werden die Variablen nach Möglichkeit in einer Übertragungseinheit zusammengefasst. Bei UDP ist eine Übertragungseinheit 256 Bytes groß. Passen nicht alle Variablen der Liste in eine Übertragungseinheit, werden für diese Liste mehrere Übertragungseinheiten gebildet. Ist die Option nicht aktiviert, kommt jede Variable in eine eigene Übertragungseinheit.

Wenn "Übertragung bei Änderung" konfiguriert ist, wird für jede Übertragungseinheit getrennt geprüft, ob sie geändert ist und gesendet werden muss.

Basisidentifier: Der Basisidentifier wird als eindeutige Kennung benutzt, um Variablenlisten verschiedener Projekte auszutauschen. Variablenlisten mit gleichem Basisidentifier werden ausgetauscht.

Es ist darauf zu achten, dass die Definitionen der Variablenlisten mit gleichem Basisidentifier in den verschiedenen Projekten übereinstimmen.

Hierzu kann das Feature "Dateiverknüpfung" verwendet werden. Aus einem Projekt wird die Variablenliste exportiert. Die anderen Projekte importieren sie, wodurch der Inhalt nicht neu eingegeben werden muss.

Damit die Daten korrekt zwischen zwei Steuerungen ausgetauscht werden, müssen die globalen Variablenlisten in den beiden Projekten übereinstimmen. Ein Projekt kann die Datei vor dem Übersetzen exportieren, die anderen Projekte sollten die vor dem Übersetzen importieren.

Neben einfachen Datentypen kann eine Variablenliste auch Strukturen und Arrays enthalten. Die Elemente dieser zusammengesetzten Datentypen werden einzeln versendet. Wenn eine Variablenliste größer als eine Übertragungseinheit ist, werden die Daten auf mehrere Übertragungseinheiten aufgeteilt. Es kann darum nicht zugesichert werden, dass alle Daten der Variablenliste in einem Zyklus empfangen werden.

Teile der Variablenliste können in verschiedenen Zyklen empfangen werden. Dies ist auch für Variablen mit Struktur- und Arraytypen möglich.

8.6 Programm generieren

Nach der Erstellung eines Projektes in CoDeSys in den Programmiersprachen nach IEC61131-3, muss das Projekt in ein für die Steuerung ausführbares Programm übersetzt werden.

Über den Menüpunkt **Online -> Einloggen** wird automatisch aus dem aktuell bearbeiteten Projekt ein ausführbares Programm erzeugt und ins RAM der Steuerung übertragen.

Besteht eine Online-Verbindung zur Steuerung, ist es über den Menüpunkt **Online -> Bootprojekt erzeugen** möglich, das aktuelle Projekt ins Flash-PROM der Steuerung zu übertragen. Zusätzlich wird auf der Festplatte im Projektverzeichnis eine Datei ”Projektname.prg” erzeugt, wenn unter **Projekt -> Optionen -> Symbolkonfiguration** das Kontrollfeld ”Binärfile der Applikation erzeugen” angewählt ist.

Diese Datei kann dann auch über die Monitorschnittstelle der Steuerung per Z-Modem-Übertragung in die Steuerung geladen werden, wobei der Dateiname des Bootprojektes in ”default.prg” geändert werden muss. Für die ETCxC kann diese Datei auch über das ETC-MMI in die Steuerung übertragen werden.

8.7 Schnittstelle zur ETC

8.7.1 Datenbausteine

Der integrierten SPS steht ein interner RAM-Bereich zur Verfügung, der in 16 sogenannte Datenbausteine unterteilt ist. Jeder Datenbaustein (DB) umfasst 256 Datenworte (DW) zu 16 Bit.

8.7.2 Systemvariablen der ETCxC

Die Systemvariablen stellen eine vordefinierte Menge von Variablen aus den Datenbausteinen dar.

In CoDeSys können die Systemvariablen über den Menüpunkt **Einfügen → Operand → Systemvariablen** oder die Eingabehilfe (über Funktionstaste ”F2”) in ein Projekt eingefügt werden.

Nachfolgend eine Liste der definierten Systemvariablen:

Datenbaustein 0

DB0_sps2nc_taste_an_funktion_w	%MW0.0	
DB0_sps2nc_nc_fehlerquittung_w	%MW0.1	
DB0_nc2sps_fehlertext_s	%MW0.16	
DB0_nc2sps_fehlerinfo_s	%MW0.96	
DB0_nc2sps_aktuelles_menu_w	%MW0.128	
DB0_nc2sps_ob20_ereignis_w	%MW0.129	
DB0_nc2sps_einzeltastenmeldung_w	%MW0.130	
DB0_nc2sps_tastennummer_b	%MB0.130.0	
DB0_nc2sps_tastenzustand_b	%MB0.130.1	
DB0_nc2sps_eingabeabschluss_w	%MW0.131	
DB0_nc2sps_funktionsstatus_w	%MW0.132	
DB0_nc2sps_nc_fehlerzaehler_w	%MW0.133	
DB0_nc2sps_nc_fehlerklasse_w	%MW0.134	
DB0_nc2sps_nc_fehlermodul_w	%MW0.135	
DB0_nc2sps_nc_fehlernummer_w	%MW0.136	
DB0_nc2sps_aktuelles_untermenu_w	%MW0.137	
DB0_nc2sps_g253_textausgabe_s	%MW0.148	

Datenbaustein 1

DB1_sps2nc_notaus_bit	%MX1.0.0	
DB1_sps2nc_vorschubfreigabe_bit	%MX1.0.1	
DB1_sps2nc_soforthalt_bit	%MX1.0.2	
DB1_sps2nc_vorschubfreigabe_w	%MW1.1	
DB1_sps2nc_position_halt_w	%MW1.2	
DB1_sps2nc_verfahrtastenfreigabe_plus_w	%MW1.3	
DB1_sps2nc_verfahrtastenfreigabe_minus_w	%MW1.4	
DB1_sps2nc_reglerfreigabe_w	%MW1.5	
DB1_sps2nc_einlesefreigabe_bit:	%MX1.7.0	
DB1_sps2nc_programmstart_b	%MB1.9.0	
DB1_sps2nc_use_spsoverride_b	%MB1.9.1	
DB1_sps2nc_spindel_ein_aus_b	%MB1.10.0	
DB1_sps2nc_spindel_richtung_b	%MB1.11.0	

DB1_sps2nc_programmstop_b	%MB1.12.0	
DB1_sps2nc_unterbrechen_bit	%MX1.12.8	
DB1_sps2nc_einzel_folgesatz_bit	%MX1.13.0	
DB1_sps2nc_satzausblenden_bit	%MX1.13.8	
DB1_sps2nc_rueckzug_bit	%MX1.14.0	
DB1_sps2nc_tastensignale_aw	%MW1.16	ARRAY[0..1] OF WORD
DB1_sps2nc_qbit_signale_aw	%MW1.18	ARRAY[0..3] OF WORD
DB1_sps2nc_freigaben_mmi_w	%MW1.31	
DB1_sps2nc_mfkt_quitt_bit	%MX1.32.0	
DB1_sps2nc_sfkt_quitt_bit	%MX1.33.0	
DB1_sps2nc_hfkt_quitt_bit	%MX1.33.0	
DB1_sps2nc_tfkt_quitt_bit	%MX1.34.0	
DB1_sps2nc_extsync_enable_bit	%MX1.35.0	
DB1_sps2nc_programmhalt_aktiv_bit	%MX1.35.8	
DB1_sps2nc_qin_mask_aw	%MW1.36	ARRAY[0..1] OF WORD
DB1_sps2nc_qout_mask_aw	%MW1.38	ARRAY[0..1] OF WORD
DB1_sps2nc_qout_akt_aw	%MW1.40	ARRAY[0..1] OF WORD
DB1_sps2nc_qin_offset_ab	%MW1.42	ARRAY[0..1] OF BYTE
DB1_sps2nc_qout_offset_ab	%MW1.43	ARRAY[0..1] OF BYTE
DB1_sps2nc_spsoverride_allg_w	%MW1.80	
DB1_sps2nc_spsoverride_spindel_w	%MW1.81	
DB1_sps2nc_spsoverride_ozillation_w	%MW1.82	
DB1_sps2nc_spsoverride_sps_w	%MW1.83	
DB1_sps2nc_verfahr_ab	%MB1.84.0	ARRAY[0..11] OF BYTE
DB1_nc2sps_betriebsbereit_1_bit	%MX1.128.0	
DB1_nc2sps_betriebsbereit_2_bit	%MX1.128.1	
DB1_nc2sps_referenzpunkt_w	%MW1.129	
DB1_nc2sps_referenzfahrt_laeuft_w	%MW1.130	
DB1_nc2sps_betriebsart_w	%MW1.131	
DB1_nc2sps_nc_programm_laeuft_bit	%MX1.132.0	
DB1_nc2sps_nc_programm_laeuft_1_bit	%MX1.132.1	
DB1_nc2sps_satzvorlauf_aktiv_bit	%MX1.132.2	
DB1_nc2sps_einzelsatz_laeuft_bit:	%MX1.132.3	
DB1_nc2sps_unterbrechen_aktiv_bit	%MX1.132.4	
DB1_nc2sps_alle_achsen_stehen_bit	%MX1.132.8	
DB1_nc2sps_wahlweise_halt_b	%MB1.133.0	
DB1_nc2sps_programmstartzaehler_b	%MB1.133.1	
DB1_nc2sps_achse_faehrt_w	%MW1.134	
DB1_nc2sps_modal_am_ziel_w	%MW1.135	
DB1_nc2sps_spindel_aktiv_w	%MW1.136	
DB1_nc2sps_programmstart_b	%MB1.137.0	
DB1_nc2sps_canmodul_vorhanden_w	%MW1.138	
DB1_nc2sps_can_status_w	%MW1.139	
DB1_nc2sps_programmstop_b	%MB1.142.0	
DB1_nc2sps_vorschub_halt_b	%MB1.144.0	
DB1_nc2sps_mfkt_strobe_bit	%MX1.150.0	
DB1_nc2sps_mfkt_w	%MW1.151	
DB1_nc2sps_hfkt_strobe_bit	%MX1.159.0	
DB1_nc2sps_hfkt_w	%MW1.160	

DB1_nc2sps_sfkt_strobe_bit	%MX1.159.0	
DB1_nc2sps_sfkt_w	%MW1.160	
DB1_nc2sps_tfkt_strobe_bit	%MX1.168.0	
DB1_nc2sps_tfkt_w	%MW1.169	
DB1_nc2sps_refpunkt_angefahren_w	%MW1.177	
DB1_nc2sps_endschalter_plus_w	%MW1.178	
DB1_nc2sps_endschalter_minus_w	%MW1.179	
DB1_nc2sps_referenznocken_w	%MW1.180	
DB1_nc2sps_reserveeingang_w	%MW1.181	
DB1_nc2sps_slave_endschalter_plus_w	%MW1.182	
DB1_nc2sps_slave_endschalter_minus_w	%MW1.183	
DB1_nc2sps_slave_referenznocken_w	%MW1.184	
DB1_nc2sps_slave_reserveeingang_w	%MW1.185	
DB1_nc2sps_copmodul_vorhanden_aw	%MW1.192	ARRAY[0..7] OF WORD
DB1_nc2sps_spsoverride_allg_w	%MW1.200	
DB1_nc2sps_spsoverride_spindel_w	%MW1.201	
DB1_nc2sps_spsoverride_ozillation_w	%MW1.202	
DB1_nc2sps_spsoverride_sps_w		%MW1.203
DB1_nc2sps_verfahr_ab	%MB1.204.0	ARRAY[0..11] OF BYTE
DB1_nc2sps_mmi_tasten_aw	%MW1.212	ARRAY[0..7] OF WORD

Datenbaustein 2

DB2_sps2mmi_zustand_aw	%MW2.0	ARRAY[0..15] OF WORD
DB2_sps2mmi_anzeige_aw	%MW2.16	ARRAY[0..63] OF WORD
DB2_sps2mmi_hinweis_aw	%MW2.80	ARRAY[0..15] OF WORD
DB2_sps2mmi_quit_cnt_w	%MW2.96	
DB2_sps2mmi_msg_cnt1_w	%MW2.97	
DB2_sps2mmi_msg_auftrag_w	%MW2.98	
DB2_sps2mmi_msg_laenge_w	%MW2.99	
DB2_sps2mmi_msg_daten_aw	%MW2.100	ARRAY[0..25] OF WORD
DB2_sps2mmi_msg_cnt2_w	%MW2.126	
DB2_sps2mmi_ok_cnt_w	%MW2.127	
DB2_mmi2sps_zustand_aw	%MW2.128	ARRAY[0..15] OF WORD
DB2_mmi2sps_tasten_aw	%MW2.144	ARRAY[0..15] OF WORD
DB2_mmi2sps_quit_cnt_w	%MW2.160	
DB2_mmi2sps_msg_cnt1_w	%MW2.161	
DB2_mmi2sps_msg_auftrag_w	%MW2.162	
DB2_mmi2sps_msg_laenge_w	%MW2.163	
DB2_mmi2sps_msg_daten_aw	%MW2.164	ARRAY[0..25] OF WORD
DB2_mmi2sps_msg_cnt2_w	%MW2.190	
DB2_mmi2sps_ok_cnt_w	%MW2.191	
DB2_nc2sps_technodaten_aw	%MW2.192	ARRAY[0..31] OF WORD
DB2_nc2sps_maschinenkonstante_aw	%MW2.224	ARRAY[0..31] OF WORD

Datenbaustein 13

DB13_nc2mmi_achskennung_ab	%MW13.12	ARRAY[0..15] OF BYTE

8.7.3 Systemvariablen der ETCxM

Zur Zeit stehen bei der ETCxM die Datenbausteine der SPS als Datenspeicher frei zur Verfügung.

8.7.4 Maschinenkonstanten in der ETCxC verwenden

In der ETCxC stehen die Maschinenkonstanten **MK_DW224_255** zur freien Verfügung, um dort maschinenspezifische Werte abzulegen.

Für die ETCxC werden diese im Datenbaustein 2 ab Datenwort 224 eingeblendet, so dass dort der Inhalt von der SPS ausgelesen werden kann. Die Systemvariable ”DB2_nc2sps_maschinenkonstante_aw” zeigt auf die Werte dieser Maschinenkonstanten.

Die Maschinenkonstanten **MK_TECHNOLOGY_DATA_1** bis **MK_TECHNOLOGY_DATA_4** stehen für technologieabhängige Daten zur Verfügung. Die Inhalte dieser Maschinenkonstanten werden im Parameterfeld abgelegt.

Index	Maschinenkonstante
760 ... 769	MK_TECHNOLOGY_DATA_1
770 ... 779	MK_TECHNOLOGY_DATA_2
780 ... 789	MK_TECHNOLOGY_DATA_3
790 ... 799	MK_TECHNOLOGY_DATA_4

Diese Parameterfeldeinträge können über die Funktionen **READ_PARAM_INT**, **READ_PARAM_DINT** usw. ausgelesen werden.

8.7.5 Maschinenkonstanten in der ETCxM verwenden

Die Maschinenkonstanten **MK_TECHNOLOGY_DATA_1** bis **MK_TECHNOLOGY_DATA_4** stehen für technologie-abhängige Daten zur Verfügung.

Diese Maschinenkonstanten enthalten jeweils 10 Einträge. Der Wertebereich jedes Eintrags entspricht einem 64-bit floating point-Wert (IEC61131-Datentyp LREAL). Die einzelnen Einträge können über die Funktion ”GetUserParam()” von der SPS ausgelesen werden.

8.7.6 Parameterfeld der ETCxC

Die ETCxC stellt ein sogenanntes Parameterfeld (P-Feld) zur Verfügung, das u. a. verschiedene Systemparameter enthält, die eventuell interessant für die SPS sind.

In der ”SysEtc.LIB” stehen Funktionen zur Verfügung, mit denen lesend und schreibend auf Parametereinträge zugegriffen werden kann (siehe ”READ_PARAM_INT” bzw. ”WRITE_PARAM_INT” usw.).

Die einzelnen Einträge in dem Parameterfeld sind im Kapitel ”ETC-Programmierung NC-Betriebssystem” beschrieben.

8.7.7 Betriebsdaten der ETCxC

Die Steuerung hat ein Betriebsdatenfeld, in dem bisher nur 2 Einträge von der Steuerung verwendet werden. Über die Funktionen READ_SYSPARAM und WRITE_SYSPARAM kann auf die Einträge zugegriffen werden.

Index	Beschreibung
101	Zykluszeit einer zweiten Task (PLC_PRG2) [10 ms]
102	Zykluszeit der SPS (PLC_PRG) [ms]

Hinweis!

Die Zykluszeit der SPS muss gleich dem Grobinterpolationstakt sein (siehe auch Maschinenkonstante MK_DELTAT).

8.7.8 Fehlermeldungen bei der ETCxM auslesen

Bei der ETCxM können aktive Fehlermeldungen über Funktionsbausteine der Bibliothek ”mc.lib” ausgelesen werden. Dabei werden achsspezifische Fehler und Betriebssystemfehler unterschieden. Es gibt jeweils eine Funktion zum Auslesen und zum Quittieren (Rücksetzen) eines anstehenden Fehlers.

Fehlerquelle	SPS-Bausteine
Achsen	MC_ReadAxisError MC_Reset
System	MC_ReadSysError MC_ResetSysError

8.8 Bibliothek

Die Funktionen und Funktionsblöcke der beiden Achsausprägungen ETCxC und ETCxM sind bis auf wenige Ausnahmen gleich. Trotzdem gibt es für jede Variante eine ETC-System-Bibliothek:

ETCxC: SysEtc.LIB

ETCxM: SysETCxM.LIB

Im Folgenden werden alle Funktionen im Einzelnen beschrieben. Diese finden sich in beiden Bibliotheken wieder. Spezielle Funktionen der Varianten sind gekennzeichnet.

8.8.1 Allgemeine Funktionen

8.8.1.1 CenterString

Deklaration

```
FUNCTION CenterString: INT
VAR_INPUT
    string_s        : STRING(255)     (* String, der umformatiert werden soll *)
    linelen_di      : DINT            (* Zeilenlänge für Zentrierung *)
END_VAR
```

Beschreibung

Die Funktion fügt Leerzeichen zur Zentrierung in den String ein. Der Returnwert entspricht der resultierenden Gesamtlänge des Strings.

Stellen Sie sicher, dass die angegebene Stringvariable groß genug ist, um den resultierenden String aufzunehmen!

Beispiel

```
test_s              : STRING (20):= 'Hallo';
CenterString(test_s, 20);
```

Durch den Aufruf von CenterString werden 7 Leerzeichen vor "Hallo" eingefügt.

8.8.1.2 FMOD

Deklaration

```
FUNCTION FMOD: LREAL
VAR_INPUT
    x: LREAL;
    y: LREAL;
END_VAR
```

Beschreibung

Die Funktion liefert den floating point Rest der Division x/y (mit Vorzeichen von x).

Beispiel

```
Ergebnis_lr         : LREAL;
Ergebnis_lr         := FMOD(10.5, 3.2); Ergebnis_lr = 0.9
```

8.8.1.3 Format

Deklaration

```
FUNCTION FORMAT: INT
VAR_INPUT
    STRING_S      : STRING(255);    (* Zielstring *)
    FORMAT_S      : STRING(80);     (* Formatstring *)
    PARAMETER_P   : DINT;           (* Adresse einer Struktur oder Variablen *)
END_VAR
```

Beschreibung

Alle Zeichen im Formatstring, die nicht zu einer Formatangabe gehören, werden in STRING_S kopiert.

Formatstring: Das %-Zeichen in einem Formatstring eröffnet eine Formatangabe mit der allgemeinen Form:

% Flag Ausgabefeldlänge Genauigkeit Datentypangabe

Flag	
<nichts>	rechtsbündig, führende Leerzeichen / Nullen
-	linksbündig, nachfolgende Leerzeichen / Nullen
+	Vorzeichen immer ausgeben

Ausgabefeldlänge	
0n	min. n Stellen, mit Nullen auffüllen
n	min. n Stellen, mit Leerzeichen auffüllen

Genauigkeit	
<nichts>	6 Stellen
.0	keinen Dezimalpunkt ausgeben
.n	max. n Nachkommastellen ausgeben

Datentyp (CoDeSys)	Datentypangabe (Formatstring)
DWORD	u
DINT	d
LREAL	lf
STRING	s

Der Returnwert entspricht der Anzahl der Zeichen, die in den String STRING_S kopiert wurden.

Beispiel

```
TYPE TEST_R
STRUCT
    Dw          : DWORD;
    L           : LREAL;
END_STRUCT
END_TYPE

    s_s         : STRING(80);
    ret_i       : INT;
    di          : DINT;
    t_r         : TEST_R;
```

Ein Parameter:

```
    di          : 345;
    ret_i       : = FORMAT(s_s, '%d', ADR(di)); ret_i = 4; s_s = >-345'
```

Mehrere Parameter:

```
    t_r.dw      : = 123;
    t_r.lr      : = 4.321;
    ret_i       : = FORMAT(s_s, '1: %u, 2: %f', ADR(t_r)); ret_i = 19; s_s = '1: 123, 2: 4.321000'
```

8.8.1.4 GetFirmwareVersion

Deklaration

```
FUNCTION GetFirmwareVersion: STRING(80)
VAR_INPUT
    type_i      : INT;
END_VAR
```

Parameter

type_i	Versionsstring
1	NCR
2	Bootloader

Beschreibung

Die Funktion liefert Versionsstrings.

8.8.1.5 GetMacAddr (nur ETCxM)

Deklaration

```
FUNCTION GetMacAddr: BOOL
VAR_INPUT
    pMac        : DINT;         (* Adresse eines Arrays vom Typ ARRAY[0..5] OF BYTE *)

END_VAR
```

Beschreibung

Mit dieser Funktion kann die MAC-Adresse des Ethernet-Controllers der Steuerung ausgelesen werden. Der Funktion muss die Adresse eines 6 Byte großen Speicherbereichs übergeben werden (s. Beispiel). Der Returnwert der Funktion signalisiert, ob die Adresse gelesen werden konnte.

Beispiel

```
ret_bit     : BOOL;
mac_a       : ARRAY[0..5] OF BYTE;
ret_bit     : = GetMacAddr(ADR(mac_ab));
```

8.8.1.6 GetUserParam (nur ETCxM)

Deklaration

```
FUNCTION GetUserParam: LREAL
VAR_INPUT
    index_dw    : DWORD;        (* Index von MK_TECHNOLOGIEDATENx *)
END_VAR
```

Beschreibung

Mit dieser Funktion können die Werte der Maschinenkonstanten MK_TECHNOLOGIEDATEN1 bis MK_TECHNOLOGIEDATEN4 gelesen werden. Der zu übergebende Wert *index_dw* entspricht der Position (beginnend mit 0) ab dem ersten Eintrag von MK_TECHNOLOGIEDATEN1 fortlaufend durchnummeriert bis zum letzten Eintrag von MK_TECHNOLOGIEDATEN4.

Beispiel

```
MK_TECHNOLOGIE-DATEN1   1.1, 0, 0, 0, 0, 0, 0, 0, 0;
MK_TECHNOLOGIE-DATEN2   0, 4.5, 3, -5.7, 0, 0, 0, 0, 0;
wert_d                  : DINT;
wert_lr                 : LREAL;
wert_lr                 : = GetUserParam(9); (* liefert den Wert -1.1 *)
wert_di                 : = LREAL_TO_DINT(GetUserParam(12)); (* liefert den Wert 3 *)
```

8.8.1.7 IO_SET

Deklaration

```
FUNCTION IO_SET: BOOL
VAR_INPUT
    byte_w      : WORD;
    bit_w       : WORD;
END_VAR
```

Beschreibung

Die Funktion invertiert den Zustand eines Ausgangs. In dem Ausgangsbyte *byte_w* (Wertebereich 0 ... 31) wird das Bit *bit_w* (Wertebereich 0 ... 7) invertiert. Der Returnwert der Funktion ist ohne Bedeutung.

Beispiel

```
IO_SET(1,3);        invertiert den Ausgang Q0.10
```

8.8.1.8 Load_Param

Deklaration

```
FUNCTION LOAD_PARAM : DINT
VAR_INPUT
    dateiname_s     : STRING(15);     (* ”device:Dateiname” *)
    pindex_di       : DINT;           (* P-Feldindex *)
    anzahl_di       : DINT;           (* Anzahl der P-Feldwerte *)
END_VAR
```

Beschreibung

LOAD_PARAM liest *anzahl_di* P-Feldwerte von der RAM-Disk (device: rd), dem FLASH-PROM (device: sd) oder einer Diskette (device: fd), die vorher mit SAVE_PARAM in der Datei gespeichert wurden, und trägt die Werte ab dem Index *pindex_di* in das P-Feld ein.

Ein Returnwert ungleich 0 signalisiert einen Fehler (📖 381).

Beispiel

```
ret_di          : DINT;
ret_di          := LOAD_PARAM('rd:daten.dat', 1200, 30);
```

Es werden 30 P-Feldwerte aus der Datei ”daten.dat” von der RAM-Disk gelesen und ab Index 1200 im P-Feld abgelegt.

8.8.1.9 READ_PARAM_INT (nur ETCxC)

Deklaration

```
FUNCTION READ_PARAM_INT: INT
VAR_INPUT
    IDX_DI          : DINT;           (* Parameterindex *)
END_VAR
```

Beschreibung

Die Funktion liefert den Wert des Parameters *idx_di* (als Datentyp INT) aus dem P-Feld.

8.8.1.10 READ_PARAM_DINT (nur ETCxC)

Deklaration

```
FUNCTION READ_PARAM_DINT: DINT
VAR_INPUT
    IDX_DI          : DINT;           (* Parameterindex *)
END_VAR
```

Beschreibung

Die Funktion liefert den Wert des Parameters *idx_di* (als Datentyp DINT) aus dem P-Feld.

8.8.1.11 READ_PARAM_REAL (nur ETCxC)

Deklaration

```
FUNCTION
VAR_INPUT
    IDX_DI          : DINT;          (* Parameterindex *)
END_VAR
```

Beschreibung

Die Funktion liefert den Wert des Parameters *idx_di* (als Datentyp REAL) aus dem P-Feld.

8.8.1.12 READ_PARAM_LREAL (nur ETCxC)

Deklaration

```
FUNCTION READ_PARAM_LREAL: LREAL
VAR_INPUT
    IDX_DI          : DINT;          (* Parameterindex *)
END_VAR
```

Beschreibung

Die Funktion liefert den Wert des Parameters *idx_di* (als Datentyp LREAL) aus dem P-Feld.

8.8.1.13 READ_SYSPARAM (nur ETCxC)

Deklaration

```
FUNCTION READ_SYSPARAM: WORD
VAR_INPUT
    IDX_I           : DINT;
END_VAR
```

Beschreibung

READ_SYSPARAM liest einen Wert aus den Betriebsdaten.

Beispiel

```
wert_w          : WORD;
wert_w          : = READ_SYSPARAM(101);
```

Liefert den Wert des Betriebsdatenworts 101.

8.8.1.14 READ_TOOLDATA (nur ETCxC)

Deklaration

```
FUNCTION READ_TOOLDATA: INT
VAR_INPUT
    tnr_dw          : DWORD;    * T (Werkzeugnummer) *)
    anzahl_dw       : DWORD;    (* Anzahl der Werkzeugdaten *)
    puffer_p        : DINT;     (* Adresse, ab der die Daten abgelegt werden sollen *)
END_VAR
```

Beschreibung

Die Funktion liest die Werkzeugdaten für das angegebene T (Werkzeug).

Beispiel

```
TYPE WERKZEUG_TR
STRUCT
    radius_lr          : LREAL; (* Werkzeugradius in mm *)
    typ_lr             : LREAL; (* Werkzeugtyp *)
    schwester_lr       : LREAL; (* Nr. des Schwesterwerkzeuges *)
    original_lr        : LREAL; (* Nr. des Originalwerkzeuges *)
    magazin_lr         : LREAL; (* Magazinplatz *)
    s_zugeordnet_lr    : LREAL; (* dem Magazinplatz zugeordnetes S *)
    s_zeit_vorgabe_lr  : LREAL; (* vorgegebene Standzeit *)
    s_zeit_aktuell_lr  : LREAL; (* aktuelle Standzeit *)
    wz_aktuell_lr      : LREAL; (* aktuelles Werkzeug *)
    drehzahl_lr        : LREAL; (* Drehzahl des Werkzeuges *)
END_STRUCT
END_TYPE
    wzdaten_tr         : WERKZEUG_TR;
    ret_i              : INT;
    ret_i              := READ_TOOLDATA(2, 10, ADR(wzdaten_tr));
```

Die Werkzeugdaten für Werkzeug 2 (T2) werden in der Strukturvariable *wzdaten_tr* abgelegt.

8.8.1.15 SAVE_PARAM (nur ETCxC)

Deklaration

```
FUNCTION SAVE_PARAM: DINT
VAR_INPUT
    dateiname_s    : STRING(15);    (* ”device:Dateiname” *)
    pindex_di      : DINT;          (* P-Feldindex *)
    anzahl_di      : DINT;          (* Anzahl der P-Feldwerte *)
END_VAR
```

Beschreibung

SAVE_PARAM speichert *anzahl_di* P-Feldwerte ab Index *pindex_di* unter dem Dateinamen ”dateiname_s” auf der RAM-Disk (device: rd), im FLASH-PROM (device: sd) oder auf einer Diskette (device: fd).

Ein Returnwert ungleich 0 signalisiert einen Fehler (📖 381).

Beispiel

```
ret_di          : DINT;
ret_di          : = SAVE_PARAM(’rd:daten.dat’, 1200, 30);
```

Es werden 30 P-Feldwerte ab Index 1200 in die Datei ”daten.dat” auf der RAM-Disk gespeichert.

8.8.1.16 SETINPUT_BIT

Deklaration

```
FUNCTION SETINPUT_BIT: BOOL
VAR_INPUT
    WORD_W      : WORD;      (* Nr. des Datenwortes *)
    BIT_W       : WORD;      (* Eingang der IO-Karte *)
END_VAR
```

Beschreibung

Die Funktion setzt ein Bit im Eingangsprozessabbild. Im Eingangsp*rozessabbild des Datenwortes word_w (Wertebereich 0 ... 63) wird Bit bit_w* (Wertebereich 0 ... 31) gesetzt. Damit kann eine Testumgebung (ohne Hardware) geschaffen werden.

Der Returnwert FALSE signalisiert einen falschen Übergabeparameter.

8.8.1.17 SETINPUT_WORD

Deklaration

```
FUNCTION SETINPUT_WORD: BOOL
VAR_INPUT
    WORD_W      : WORD;      (* Nr. des Datenwortes *)
    VAL_W       : WORD;
END_VAR
```

Beschreibung

Die Funktion führt eine boolsche ODER-Verknüpfung mit dem Wert *val_w* und dem Datenwort *word_w* des Eingangsprozessabbildes durch und schreibt das Ergebnis in das selbe Datenwort.

Returnwert FALSE signalisiert einen falschen Übergabeparameter.

Die Funktion ermöglicht es einem Entwickler eine Testumgebung (ohne Hardware) zu schaffen.

8.8.1.18 SINGLEBLOCK (nur ETCxC)

Deklaration

```
FUNCTION SINGLEBLOCK: BOOL
VAR_INPUT
    Satzstring      : STRING(80);    (* Satzstring *)
    parameter_p     : DINT           (* Adresse einer Struktur *)
END_VAR
```

Beschreibung

Diese Funktion sendet einen Einzelsatz an den NC-Rechner. (Der Satzstring ist zu verwenden, wie unter SPSERROR beschrieben.)

Beispiel 1

```
SINGLEBLOCK('G0 X100.2 Y50.73', 0);
```

Ein Einzelsatz ohne Parameter wird an den NC-Rechner gesendet.

Beispiel 2

```
TYPE PARAMETER_TR
STRUCT
    pos_X_lr        : LREAL;
    pos_Y_di        : DINT;
END_STRUCT
END_TYPE
param_tr:           PARAMETER_TR;
SINGLEBLOCK(>G0 X%f Y%d', ADR(param_tr));
```

Ein Einzelsatz mit Parametern wird an den NC-Rechner gesendet.

8.8.1.19 SPSERROR

Deklaration

```
FUNCTION SPSERROR: BOOL
VAR_INPUT
    fehler_di       : DINT;         (* Fehlernummer *)
    klasse_dw       : DWORD;        (* Fehlerklasse *)
    formatstring    : STRING(80);   (* Formatstring *)
    parameter_p     : DINT;         (* Adr. einer Struktur, die die Parameter beinhaltet *)
END_VAR
```

Beschreibung

Die Funktion setzt eine Fehlermeldung ab, die auf einem angeschlossenen Bedienteil zur Anzeige gebracht wird. Die Fehlernummer *fehler_di* dient gleichzeitig als Referenz auf einen Fehlertext. Die Fehlertexte können auf einem angeschlossenen PC (für beide ETC Varianten) oder auf dem ETC-MMI (nur ETCxC) angezeigt werden.

Die Fehlerklassen *klasse_dw* sind folgendermaßen unterteilt:

1: lokal leichter Fehler

2: lokal schwerer Fehler

3: global leichter Fehler

4: global schwerer Fehler

Die Fehlerklassen 3 und 4 führen zum Abbruch eines Programms bzw. einer Achsbewegung.

Zu dem über die Fehlernummer referenzierten Fehlertext kann eine zusätzliche Information ausgegeben werden. Über den Formatstring können weiterer Text bzw. numerische Werte übergeben werden.

Formatstring: Das %-Zeichen in einem Formatstring eröffnet eine Formatangabe mit der allgemeinen Form:

% Flag Ausgabefeldlänge Genauigkeit Datentypangabe

Flag	
<nichts>	rechtsbündig, führende Leerzeichen / Nullen
-	linksbündig, nachfolgende Leerzeichen / Nullen
+	Vorzeichen immer ausgeben

Ausgabefeldlänge	
0n	min. n Stellen, mit Nullen auffüllen
n	min. n Stellen, mit Leerzeichen auffüllen

Genauigkeit	
<nichts>	6 Stellen
.0	keinen Dezimalpunkt ausgeben
.n	max. n Nachkommastellen ausgeben

Datentyp (CoDeSys)	**Datentypangabe (Formatstring)**
DWORD	u
DINT	d
REAL	f e
LREAL	lf le
STRING	s

Der Returnwert der Funktion ist ohne Bedeutung.

Beispiel 1

```
SPSERROR(304, 4, 'Spindel läuft nicht an', 0);
```

Beispiel 2

```
TYPE PARAMETER_TR
STRUCT
    auftragsnr_dw       : DWORD;
    text_ps             : POINTER TO STRING;
    pos_lr              : LREAL;
END_STRUCT
END_TYPE
param_tr
string_s
param_tr.auftragsnr_dw
param_tr.text_ps
param_tr.pos_lr
SPSERROR(303, 1, 'Auftragsnr%u %s %9.3lf', ADR(param_tr));
```

8.8.1.20 SETLANGUAGE (nur ETCxM)

Deklaration

```
FUNCTION SETLANGUAGE: BOOL
VAR_INPUT
    Sprache         : INT;
END_VAR
```

Beschreibung

Mit dieser Funktion wird zwischen verschiedenen Sprachen für Ausgabetexte umgeschaltet. Mit Sprache = 0 werden die Ausgabetextdateien ncrspch0.txt/spsspch0.txt, mit Sprache = 1 die Dateien ncrspch1.txt/ spsspch1.txt usw. angewählt.

Der Returnwert der Funktion ist ohne Bedeutung.

Beispiel

```
SETLANGUAGE(4);
```

Ausgabetextdateien ncrspch4.txt und spsspch4.txt werden ausgewählt.

8.8.1.21 STRTOF

Deklaration

```
FUNCTION STRTOF: REAL
VAR_INPUT
    SString         : STRING(255);
    PIndex          : POINTER TO INT;
END_VAR
```

Beschreibung

Die Funktion konvertiert die Zeichen in *sString* ab der Position *Index* in einen REAL. Index enthält nach der Ausführung die Einleseendposition.

8.8.1.22 STRTOL10

Deklaration

```
FUNCTION STRTOL10: DINT
VAR_INPUT
    sString         : STRING(255);
    pIndex          : POINTER TO INT;
END_VAR
```

Beschreibung

Die Funktion konvertiert die Zeichen in *sString* ab der Position *Index* in einen DINT. Index enthält nach der Ausführung die Einleseendposition.

8.8.1.23 SYSERROR

Deklaration

```
FUNCTION SYSERROR: BOOL
VAR_INPUT
    fehler_di       : DINT;         (* Fehlernummer *)
    klasse_dw       : DWORD;        (* Fehlerklasse *)
    formatstring    : STRING(80);   (* Formatstring *)
    parameter_p     : DINT;         (* Adr. einer Struktur, die die Parameter beinhaltet *)
END_VAR
```

Beschreibung

Die Funktion SYSERROR wird wie die Funktion SPSERROR verwendet.

Mit dieser Funktion können steuerungsinterne Fehler zur Anzeige gebracht werden.

8.8.1.24 TRACE

Deklaration

```
FUNCTION TRACE: BOOL
VAR_INPUT
    Idx             : DINT;         (* Traceindex *)
    Formatstring    STRING(80);     (* Formatstring *)
    parameter_p     : DINT;         (* Adresse einer Struktur *)
END_VAR
```

Beschreibung

Die Trace-Funktion kann sehr hilfreich beim Debuggen von dynamischen Vorgängen sein. Jeder Trace kann im Diagnose-Monitor gezielt aktiviert bzw. deaktiviert werden. Dazu wird beim Aufruf der Trace-Funktion eine Nummer angegeben, anhand derer der Trace identifiziert wird. Die Tracenummern *idx* 100 ... 200 sind für die SPS reserviert, alle anderen bleiben dem NC-Rechner vorbehalten.

Der Trace-Puffer im NC-Rechner fasst 512-Trace-Meldungen. Werden die Trace-Meldungen schneller in den Puffer eingetragen als sie über die Diagnose-Schnittstelle (RS232) gesendet werden können, kann der Puffer überlaufen und Meldungen verloren gehen.

Der Returnwert der Funktion ist ohne Bedeutung.

Beispiel

```
TYPE PARAMETER_TR
STRUCT
    param1_r            : REAL;
    param2_di           : DINT;
END_STRUCT
END_TYPE
param_tr:               PARAMETER_TR;
parameter_tr.param1_r   : = 30.2
parameter_tr.param2_di  : = 20;
TRACE(100, 'REAL:%f DINT:%d', ADR(param_tr));
```

8.8.1.25 WordWrap

Deklaration

```
FUNCTION WordWrap: INT
VAR_INPUT
    string_s        : STRING(255);      (* String, der umformatiert werden soll *)
    linelen_di      : DINT;             (* maximale Zeilenlänge *)
END_VAR
```

Beschreibung

Die Funktion fügt automatisch einen Zeilenumbruch (Linefeeds) in den String ein, wenn die Zeilenlänge den angegebenen Wert überschreitet. Der Returnwert ist die resultierende Gesamtlänge des Strings.

Stellen Sie sicher, dass die angegebene Stringvariable groß genug ist, um den resultierenden String aufzunehmen!

8.8.1.26 WRITE_PARAM_INT (nur ETCxC)

Deklaration

```
FUNCTION WRITE_PARAM_INT: BOOL
VAR_INPUT
    IDX_DI          : DINT;             (* Parameterindex *)
    VAL_I           : INT;              (* Wert *)
END_VAR
```

Beschreibung

Die Funktion schreibt den Wert *val_i* an die Stelle *idx_di* ins P-Feld.

Der Returnwert FALSE signalisiert einen Fehler.

8.8.1.27 WRITE_PARAM_DINT (nur ETCxC)

Deklaration

```
FUNCTION WRITE_PARAM_DINT: BOOL
VAR_INPUT
    IDX_DI          : DINT;             (* Parameterindex *)
    VAL_DI          : DINT;             * Wert *)
END_VAR
```

Beschreibung

Die Funktion schreibt den Wert *val_i* an die Stelle *idx_di* ins P-Feld.

Der Returnwert FALSE signalisiert einen Fehler.

8.8.1.28 WRITE_PARAM_REAL (nur ETCxC)

Deklaration

```
FUNCTION WRITE_PARAM_REAL: BOOL
VAR_INPUT
    IDX_DI      : DINT;     (* Parameterindex *)
    VAL_R       : REAL;     (* Wert *)
END_VAR
```

Beschreibung

Die Funktion schreibt den Wert *val_i* an die Stelle *idx_di* ins P-Feld. Der Returnwert FALSE signalisiert einen Fehler.

8.8.1.29 WRITE_PARAM_LREAL (nur ETCxC)

Deklaration

```
FUNCTION WRITE_PARAM_LREAL: BOOL
VAR_INPUT
    IDX_DI      : DINT;     (* Parameterindex *)
    VAL_LR      : LREAL;    (* Wert *)
END_VAR
```

Beschreibung

Die Funktion schreibt den Wert *val_i* an die Stelle *idx_di* ins P-Feld.

Der Returnwert FALSE signalisiert einen Fehler.

8.8.1.30 WRITE_SYSPARAM (nur ETCxC)

Deklaration

```
FUNCTION WRITE_SYSPARAM: BOOL
VAR_INPUT
    IDX_I       : INT;      (* 0 ... 127 *)
    VAL_W       : WORD;
END_VAR
```

Beschreibung

WRITE_SYSPARAM schreibt den Wert *val_w* an die Stelle *idx_i* in die Betriebsdaten.

Returnwert 0 bedeutet falscher Index.

Beispiel

```
ret_bit     : BOOL;
et_bit      := WRITE_SYSPARAM(101, 100);
```

Schreibt den Wert 100 in das Betriebsdatenwort 101. Damit wird festgelegt, dass das PROGRAMM PLC_PRG2 (falls vorhanden) zyklisch alle 100 * 10 ms aufgerufen wird.

8.8.1.31 Returncodes der Funktionen LOAD, SAVE, LOAD_PARAM, SAVE_PARAM

Returncode	Beschreibung
-1	EEPROM defekt
-2	Checksummenfehler im EEPROM
-3	EEPROM voll
-4	Datei ist bereits geöffnet, Zugriff verweigert
-5	zu viele gleichzeitig geöffnete Dateien
-6	EEPROM Inhaltsverzeichnis ist voll
-7	Datei nicht vorhanden
-93	Checksummenfehler in der RAM-Disk
-94	nicht genug Speicher zum Anlegen der File-Buffer
-95	RAM-Disk Inhaltsverzeichnis ist voll
-96	RAM-Disk ist voll
-110	Unbekannter Devicename
-111	V24-Schnittstelle zum Diskettenlaufwerk ist belegt
-112	Diskettenlaufwerk ist nicht frei
-113	Timeout beim Zugriff auf Diskettenlaufwerk
-114	Fehler beim Zugriff auf Diskettenlaufwerk
-115	Diskettenlaufwerk meldet Übertragungsfehler
-116	Diskettenlaufwerk meldet V24-Overrun
-117	Diskette ist schreibgeschützt
-118	keine Diskette eingelegt
-119	Schreib-/Lesefehler auf der Diskette
-121	Dateiname enthält unzulässige Zeichen

Speziell für READ_PARAM, WRITE_PARAM

Returncode	Beschreibung
1	P-Feldindex außerhalb des erlaubten Bereichs

8.8.2 V24-Funktionen

8.8.2.1 ALLOCV24

Deklaration

```
FUNCTION AllocV24: DINT
VAR_INPUT
    unit_di     : DINT;     (* Schnittstelle: COM1 (X3), COM2 (X4) *)
    pri_di      : DINT;     (* Priorität (-128 ... +127) *)
END_VAR
```

Beschreibung

Eine V24-Schnittstelle muss vor Benutzung mit AllocV24() allociert werden. Danach kann sie mit InitV24() nach eigenen Bedürfnissen initialisiert werden.

Die Anforderung wird nur bedient, wenn die Schnittstelle frei ist oder die Priorität der Anforderung höher als die aktuelle Priorität ist. Eine allocierte V24-Schnittstelle kann mittels FreeV24() wieder freigegeben werden, damit wird der letzte Zustand vor der Allocation wieder hergestellt.

Mögliche Schnittstellen sind COM1 (*unit_di* = 0) und COM2 (*unit_di* = 1). Für die Priorität können Werte zwischen -128 (niedrigste) und +127 (höchste) verwendet werden, wobei eine Priorität von 127 bedeutet, dass die Schnittstelle nicht ”gestohlen” werden kann.

Die Funktion liefert die Adresse der allocierten V24-Request Struktur zurück. Wird 0 zurück geliefert konnte die Schnittstelle nicht allociert werden.

Beispiel

```
request_p       : DINT;
request_p       : = ALLOCV24(0, 110);
```

Die erste V24-Schnittstelle (X3-Stecker) wird mit Priorität 110 allociert.

8.8.2.2 INITV24

Deklaration

```
FUNCTION InitV24 : DINT
VAR_INPUT
    req_pr      : DINT;     (* Adresse der V24-Request Struktur *)
    mode_dw     : DWORD;
    flags_dw    : DWORD;
END_VAR
```

Beschreibung

Diese Funktion dient zur Einstellung der Schnittstellenparameter einer mit AllocV24 allocierten V24-Schnittstelle.

► *mode_dw* ergibt sich aus der Veroderung von Mode-Bits (📖 381).

► *flags_dw* entsteht durch Veroderung von Flag-Bits (📖 381).

Eine Returnwert ungleich 0 signalisiert einen Fehler (📖 381).

Beispiel

```
ret_di          : DINT;
requestV24_p    : DINT;
requestV24_p    : = ALLOCV24(1, 127);
ret_di          : = InitV24(requestV24_p,
                  mode_no_parity OR mode_8_bits_per_char OR
                  mode_no_rts_cts_control OR mode_stop_bit_len_2 OR
                  mode_rx_baudrate_9600 OR mode_tx_baudrate_9600 OR
                  mode_rx_buffersize_2k OR mode_tx_buffersize_2k,flags_no_protocol);
```

8.8.2.3 FREEV24

Deklaration

```
FUNCTION FreeV2: BOOL
VAR_INPUT
    req_pr      : DINT;     (* Adresse der V24-Request Struktur *)
END_VAR
```

Beschreibung

Mit dieser Funktion werden die Zugriffsrechte auf die allocierte V24-Schnittstelle wieder abgegeben und die Schnittstelle wieder in den ursprünglichen Zustand versetzt.

Der Returnwert der Funktion ist ohne Bedeutung.

Beispiel

```
requestV24_p    : DINT;
requestV24_p    : = ALLOCV24(1, 127);
REEV24(requestV24_p);
```

8.8.2.4 READBLOCKV24

Deklaration

```
FUNCTION ReadBlockV24: DINT
VAR_INPUT
    pRequest    : DINT;     (* Adresse der V24-Request Struktur *)
    pBuffer     : DINT;     (* Adresse eines Datenpuffer *)
    BufSize     : INT;      (* Größe des Puffer in Byte *)
END_VAR
```

Beschreibung

Die Funktion liest maximal *BufSize* Zeichen in den angegebenen Datenpuffer *pBuffer* aus dem V24 Empfangspuffer.

Die Funktion liefert die Anzahl der aus dem Empfangspuffer gelesenen Zeichen.

8.8.2.5 READV24

Deklaration

```
FUNCTION ReadV24: DINT
VAR_INPUT
    req_pr          : DINT;          (* Adresse der V24-Request Struktur *)
END_VAR
```

Beschreibung

Ein Zeichen aus dem Empfangspuffer lesen. Der Empfangspuffer ist als FIFO implementiert, es wird immer zeichenweise gelesen und geschrieben.

Diese Funktion kehrt in jedem Fall sofort zurück. Returnwerte kleiner 0 zeigen einen Fehler an (📖 381). Wenn das FIFO leer ist, wird -1 (EOF) zurückgegeben.

Beispiel

```
requestV24_p        : DINT;
zeichen_di          : DINT;
zeichen_di          : = READV24(requestV24_p);
```

8.8.2.6 WRITEBLOCKV24

Deklaration

```
FUNCTION WriteBlockV24: DINT
VAR_INPUT
    pRequest        : DINT;          (* Adresse der V24-Request Struktur *)
    pBuffer         : DINT;          (* Adresse eines Datenpuffer *)
    BufSize         : INT;           (* Anzahl der zu schreibenden Zeichen *)
END_VAR
```

Beschreibung

Die Funktion versucht *BufSize* Zeichen von der angegebenen Adresse *pBuffer* in den V24-Sendepuffer einzutragen. Sofern genug Platz im Sendepuffer ist, werden alle *BufSize*-Zeichen eingetragen.

Die Funktion liefert die Anzahl der in den Sendepuffer geschriebenen Zeichen zurück.

8.8.2.7 WRITEV24

Deklaration

```
FUNCTION WriteV24: DINT
VAR_INPUT
    req_pr          : DINT;          (* Adresse der V24-Request Struktur *)
    chr_di          : DINT;          (* zu schreibendes Zeichen *)
END_VAR
```

Beschreibung

Ein Zeichen in den Sendepuffer schreiben. Der Sendepuffer ist als FIFO implementiert, es wird immer zeichenweise gelesen und geschrieben.

Diese Funktion kehrt in jedem Fall sofort zurück, auch wenn das FIFO voll ist. In diesem Fall wird das Zeichen nicht geschrieben und -1 (EOF) zurückgegeben. Generell zeigt ein Returnwert kleiner 0 einen Fehler an (📖 381).

Beispiel

```
requestV24_p        : DINT;
zeichen_di          : DINT;
ret_di              : DINT;

zeichen_di          : = 65;
ret_di              : = WRITEV24(requestV24_p, zeichen_di);
```

8.8.2.8 CLRRXBUFFER

Deklaration

```
FUNCTION ClrRxBuffer: BOOL
VAR_INPUT
    req_pr          : DINT;          (* Adresse der V24-Request Struktur *)
END_VAR
```

Beschreibung

Mit dieser Funktion kann das Empfangs-FIFO komplett gelöscht werden.

Der Returnwert der Funktion ist ohne Bedeutung.

Beispiel

```
requestV24_p        : DINT;
requestV24_p        : = ALLOCV24(1, 127);
CLRRXBUFFER(requestV24_p);
```

8.8.2.9 CLRTXBUFFER

Deklaration

```
FUNCTION ClrTxBuffer: BOOL
VAR_INPUT
    req_pr          : DINT;          (* Adresse der V24-Request Struktur *)
END_VAR
```

Beschreibung

Mit dieser Funktion kann das Sender-FIFO gelöscht werden.

Der Returnwert der Funktion ist ohne Bedeutung.

Beispiel

```
requestV24_p        : DINT;
requestV24_p        : = ALLOCV24(1, 127);
CLRTXBUFFER(requestV24_p);
```

8.8.2.10 Globale Konstanten zu V24 Funktionen

(* Fehlercodes: *)	
V24_WRONG_UNIT_KDI	: DINT := -100; (* falsche V24-Schnittstellennr. *)
V24_OUTOFMEMORY_KDI	: DINT := -101; (* nicht genug RAM zum Anlegen der V24-FIFO-Puffer*)
V24_OVERRUN_ERROR_KDI	: DINT := -102; (* Overrun *)
V24_PARITY_ERROR_KDI	: DINT := -103; (* Parity-Fehler *)
V24_FRAMING_ERROR_KDI	: DINT := -104; (* Framing-Fehler *)
V24_RECEIVED_BREAK_KDI	: DINT := -105; (* Received break *)

(* Mode-Bits: *)	
MODE_EVEN_PARITY_KDW	: DWORD :=16#00000000;
MODE_ODD_PARITY_KDW	: DWORD := 16#00000004;
MODE_NO_PARITY_KDW	: DWORD := 16#00000010;
MODE_7_BITS_PER_CHAR_KDW	: DWORD := 16#00000000;
MODE_8_BITS_PER_CHAR_KDW	: DWORD := 16#00000001;
MODE_RTS_CTS_CONTROL_KDW	: DWORD := 16#00001080;
MODE_NO_RTS_CTS_CONTROL_KDW	: DWORD := 16#00000000;
MODE_STOP_BIT_LEN_1_KDW	: DWORD := 16#00000000;
MODE_STOP_BIT_LEN_2_KDW	: DWORD := 16#00000800;
MODE_BAUDRATE_2400_KDW	: DWORD := 16#00080000;
MODE_BAUDRATE_4800_KDW	: DWORD := 16#00090000;
MODE_BAUDRATE_9600_KDW	: DWORD := 16#000B0000;
MODE_BAUDRATE_19200_KDW	: DWORD := 16#000C0000;
MODE_BAUDRATE_38400_KDW	: DWORD := 16#000E0000;
MODE_BAUDRATE_57600_KDW	: DWORD := 16#000D0000;
MODE_BAUDRATE_115200_KDW	: DWORD := 16#000F0000;
MODE_RX_BUFFERSIZE_1K_KDW	: DWORD := 16#0A000000;
MODE_RX_BUFFERSIZE_2K_KDW	: DWORD := 16#0B000000;
MODE_RX_BUFFERSIZE_3K_KDW	: DWORD := 16#0C000000;
MODE_RX_BUFFERSIZE_4K_KDW	: DWORD := 16#0D000000;
MODE_TX_BUFFERSIZE_1K_KDW	: DWORD := 16#A0000000;
MODE_TX_BUFFERSIZE_2K_KDW	: DWORD := 16#B0000000;
MODE_TX_BUFFERSIZE_3K_KDW	: DWORD := 16#C0000000;
MODE_TX_BUFFERSIZE_4K_KDW	: DWORD := 16#D0000000;
(* Flag-Bits: *)	
FLAGS_NO_PROTOCOL_KDW	: DWORD := 16#00000000;
FLAGS_PLC_PROTOCOL_KDW	: DWORD := 16#00000001;
FLAGS_XON_XOFF_PROTOCOL_KDW	: DWORD := 16#00000002;
FLAGS_NAK_ON_RX_ERROR_KDW	: DWORD := 16#00000100;
FLAGS_MELD_RX_ERROR_KDW	: DWORD := 16#00000200;
FLAGS_USER_POLLING_KDW	: DWORD := 16#00001000;
FLAGS_CHANNEL_STOLEN_KDW	: DWORD := 16#80000000;

8.8.3 FILE IO-Funktionen

8.8.3.1 LOAD

Deklaration

```
FUNCTION LOAD: DINT
VAR_INPUT
    Name          : STRING(15);   (* 'device:Dateiname' *)
    daten_paten_p : DINT;         (* Adr. ab der die Daten gespeichert werden sollen *)
    len           : INT;          (* Länge der Daten in Bytes *)
END_VAR
```

Beschreibung

Mit dieser Funktion werden Daten von einem Gerät gelesen (📖 381).

Aus der Datei *name* werden von dem angegebenen Gerät *len* Bytes gelesen und ab der Speicheradresse *daten_p* abgelegt. Ohne Device-Angabe wird versucht, die Datei vom FLASHPROM zu lesen.

Ein Returnwert kleiner 0 signalisiert einen Fehler (📖 381).

Beispiel

```
werte_adi     : ARRAY[0..5] OF DINT;
anzahl_di     : DINT;
anzahl_di     := LOAD('RD:daten.dat', ADR(werte_adi), 24);
```

Sechs DINT-Werte aus der Datei "daten.dat" werden von der RAM-Disk gelesen und im *Array werte_di* abgelegt.

8.8.3.2 SAVE

Deklaration

```
FUNCTION SAVE: DINT
VAR_INPUT
    name          : STRING(15);   (* 'device:Dateiname' *)
    daten_p       : DINT;         (* Adr. ab der die Daten gespeichert werden sollen *)
    len           : INT;          (* Länge der Daten in Bytes *)
END_VAR
```

Beschreibung

Mit dieser Funktion können Daten auf einem Gerät (📖 381) gespeichert werden.

Unter *name* werden *len* Bytes ab der Speicheradresse *daten_p* auf dem angegebenen Gerät (device) gespeichert. Ohne device-Angabe wird die Datei ins FLASH-PROM geschrieben.

Ein Returnwert kleiner 0 zeigt einen Fehler beim Speichern an (📖 381).

Beispiel

```
werte_adi     : ARRAY[0..5] OF DINT;
anzahl_di     : DINT;
anzahl_di     := SAVE('RD:daten.dat', ADR(werte_adi), 24);
```

Sechs DINT-Werte aus dem Array *werte_di* werden in der Datei "daten.dat" auf der RAM-Disk gespeichert.

8.8.3.3 SetCurrentPath

Deklaration

```
FUNCTION SetCurrentPath: DINT
VAR_INPUT
    path_s          : STRING(40);       (* Gerätename und Pfad *)
END_VAR
```

Beschreibung

Legt das aktuelle Default-Laufwerk und den Verzeichnispfad für die File-IO Funktionen der Gerätetreiber (📖 381) fest. Mit dem Verzeichnispfad kann die IP-Adresse des CNC-Data Servers (Net-Disk) festgelegt werden.

Returnwert: Die Funktion liefert einen Fehlercode zurück (📖 381).

Beispiel

```
SetCurrentPath('ND:172.16.5.66');
```

8.8.3.4 SYSOPENFILE

Deklaration

```
FUNCTION SYSOPENFILE : DINT
VAR_INPUT
    filename_s      : STRING(15);
    accessmode_di   : DINT;
END_VAR
```

Beschreibung

Die Funktion dient zum Öffnen einer Datei auf dem angegebenen Gerät (📖 381). Sie liefert ein Filehandle, das für die Funktionen SysReadFile() und SysWriteFile() benötigt wird. Es können insgesamt 4 Dateien gleichzeitig geöffnet werden.

Parameter

filename_s:	Name der zu öffnenden Datei. Optional mit vorangehender Devicekennung und Doppelpunkt.
accessmode_di:	Zugriffsmode O_RDONLY oder O_WRONLY (in gobalen Konstanten definiert)

Returnwert: Filehandle (Größer oder gleich 0, wenn erfolgreich, kleiner 0 bei einem Fehler (📖 381).

Beispiel

```
handle_di       : DINT;
handle_di       := SYSOPENFILE('RD:daten.dat', O_RDONLY);
```

8.8.3.5 SYSCLOSEFILE

Deklaration

```
FUNCTION SYSCLOSEFILE: DINT
VAR_INPUT
    handle_di       : DINT;
END_VAR
```

Beschreibung

Die Funktion dient zum Schließen einer zuvor mit SysOpenFile() geöffneten Datei. Das Inhaltsverzeichnis des FLASH-PROMs wird erst im FLASH-PROM upgedatet, wenn die letzte zum Schreiben geöffnete Datei geschlossen wird. Der Returnwert der Funktion ist kleiner 0 bei einem Fehler (alle Fehlercodes sind in den globalen Konstanten der Bibliothek definiert).

Parameter

Handle_di	Filehandle (von SysOpenFile() geliefert wurde)

8.8.3.6 SYSREADFILE

Deklaration

```
FUNCTION SYSREADFILE: DINT
VAR_INPUT
    handle_di     : DINT;   (* Filehandle, das von SysOpenFile geliefert wurde *)
    buffer_p      : DINT;   (* Adresse eines Puffer, in dem die Daten abgelegt werden *)
    maxlen_di     : DINT;   (* Länge des Puffers in Byte *)
END_VAR
```

Beschreibung

Die Funktion dient zum blockweisen Lesen (sequenziell) aus einer zuvor mit SysOpenFile() geöffneten Datei. Dabei werden maximal *maxlen_di* Zeichen aus der Datei in den mit *buffer_p* angegebenen Puffer gelesen. Weniger als *maxlen_di* Zeichen werden nur gelesen, wenn vorher das Dateiende erreicht wird oder ein Fehler beim Lesen auftritt.

Returnwert: Tatsächlich gelesene Anzahl von Zeichen; 0 bei Dateiende oder kleiner 0 bei einem Fehler (📖 381).

Beispiel

```
handle_di       : DINT;
anz_di          : DINT;
ret_di          : DINT;
buffer_ab       : ARRAY[0..100] OF BYTE;

handle_di       := SYSOPENFILE('test.dat', O_RDONLY);
anz_di          := SYSREADFILE(handle_di, ADR(buffer_ab), 20);
ret_di          := SYSCLOSEFILE(handle_di);
```

8.8.3.7 SYSREADLINE

Deklaration

```
FUNCTION SYSREADLINE: DINT
VAR_INPUT
    handle_di     : DINT;   (* Filehandle, das von SysOpenFile geliefert wurde *)
    buffer_p      : DINT;   (* Adresse eines Puffer, in dem die Daten abgelegt werden *)
    maxlen_di     : DINT;   (* Länge des Puffers in Byte *)
END_VAR
```

Beschreibung

Die Funktion dient zum zeilenweisen Lesen (sequenziell) aus einer zuvor mit SysOpenFile() geöffneten Datei. Dabei wird jeweils bis zum Zeilenende aber maximal *maxlen_di* Zeichen gelesen. Die Zeile wird nur nullterminiert, falls weniger als *maxlen_d* Zeichen gelesen wurden. Das Linefeed am Ende der Zeile wird mitgelesen.

Returnwert: Tatsächlich gelesene Anzahl von Zeichen, 0 bei Dateiende oder kleiner 0 bei einem Fehler (📖 381).

8.8.3.8 SYSWRITEFILE

Deklaration

```
FUNCTION SYSWRITEFILE: DINT
VAR_INPUT
    handle_di       : DINT;     (* Filehandle, das von SysOpenFile geliefert wurde *)
    buffer_p        : DINT;     (* Adresse eines Puffer *)
    len_di          : DINT;     (* Anzahl der zuschreibenden Zeichen *)
END_VAR
```

Beschreibung

Die Funktion dient zum blockweisen Schreiben (sequenziell) in eine zuvor mit SysOpenFile() geöffnete Datei. Dabei werden *len_di* Zeichen in die Datei geschrieben. Wenn beim Schreiben ein Fehler auftritt, wird die Datei gelöscht und ein Wert kleiner 0 zurückgegeben.

Returnwert: Anzahl der geschriebenen Daten, oder kleiner 0 bei einem Fehler (📖 381).

Beispiel

```
handle_di           : DINT;
anz_di              : DINT;
ret_di              : DINT;
buffer_s            : STRING(10) := 'Hallo';

handle_di           := SYSOPENFILE('test.dat', O_WRONLY);
anz_di              := SYSWRITEFILE(handle_di, ADR(buffer_s), 5);
ret_di              := SYSCLOSEFILE(handle_di);
```

8.8.3.9 SYSREMOVEFILE

Deklaration

```
FUNCTION SYSREMOVEFILE: DINT
VAR_INPUT
    filename_s      : STRING(15);       (* Dateiname *)
END_VAR
```

Beschreibung

Die Funktion dient zum Löschen einer Datei aus dem FLASH-PROM, der RAM-Disk oder der FLOPPY.

Parameter

filename_pc:	Name der zu löschenden Datei. Optional mit vorangehender Device-Kennung und Doppelpunkt 📖 381.

8.8.3.10 SYSDISKFORMAT

Deklaration

```
FUNCTION SYSDISKFORMAT: DINT
VAR_INPUT
    device_s        : STRING(15);
END_VAR
```

Beschreibung

Mit der Funktion wird das FLASH-PROM, die RAM-Disk bzw. die FLOPPY formatiert. Dabei werden alle Daten gelöscht!

Parameter

device_pc:	Devicename 📖 381. Leerstring (" ") für das Default-Device (FLASH-PROM).

Returnwert: Kleiner 0 bei einem Fehler (alle Fehler sind in den globalen Konstanten der Bibliothek definiert).

8.8.3.11 SYSFIRSTFILE, SYSNEXTFILE

Deklaration

```
FUNCTION SYSFISRTFILE: DINT
VAR_INPUT
    info_p          : DINT;          (* Adr. einer Variablen vom Typ FILEINFO_TR *)
    pattern_s       : STRING(15);    (* Suchpattern *)
END_VAR
```

```
FUNCTION SYSNEXTFILE: DINT
VAR_INPUT
    info_p          : DINT;          (* Adr. einer Variablen vom Typ FILEINFO_TR *)
    pattern_s       : STRING(15);    (* Suchpattern *)
END_VAR
```

Beschreibung

Diese Funktionen werden benötigt um das Inhaltsverzeichnis des FLASH-PROMs, der RAM-Disk bzw. der FLOPPY auszulesen. Mit SysFirstFile() wird die angegebene Fileinfo-Struktur initialisiert und Informationen zur ersten Datei in der Struktur abgelegt. Alle folgenden Aufrufe von SysNextFile(), mit derselben Fileinfo-Struktur, legen die Informationen, der jeweils nächsten Datei in der Struktur ab.

Die Fileinfo-Struktur ist folgendermaßen aufgebaut:

```
TYPE FILEINFO_TR
STRUCT
    filename_s      : STRING(15); (* Dateiname *)
    filesize_dw     : DWORD; (* Dateigröße in Byte *)
    filenum_di      : DINT; (* Nr. des Directory-Eintrags *)
END_STRUCT
END_TYPE
```

Parameter

info_p:	Zeiger auf die Fileinfo-Struktur, in der Dateiname und Dateilänge gespeichert werden sollen.
pattern_s:	Suchpattern mit MS-DOS-Wildcards. Es wird jeweils die nächste Datei geliefert, deren Dateiname zum Pattern passt.

Returnwert: 0, wenn ok. DSK_FILE_NOT_EXIST_KDI, wenn keine weitere Datei mehr vorhanden. Alle anderen Werte signalisieren einen Fehler (alle Fehler sind in den globalen Konstanten der Bibliothek definiert).

Beispiel

```
info_st          : FILEINFO_TR;
err_di           DINT;

err_di = SYSFIRSTFILE(ADR(info_st), 'SD:*.DIN'); (* Info zum ersten File holen *)
err_di = SYSNEXTFILE(ADR(info_st), 'SD:*.DIN'); (* Info zum nächsten File holen *)
```

8.8.3.12 SYSDISKINFO

Deklaration

```
FUNCTION SYSDISKINFO: DINT
VAR_INPUT
    info_p          : DINT;          (* Adresse einer Variablen vom Typ DISKINFO_TR *)
    pattern_s       : STRING(15);    (* Suchpattern *)
END_VAR
```

Beschreibung

Die Funktion ermittelt die Belegungsdaten eines Devices und legt sie in der angegebenen Diskinfo-Struktur ab.

Die Diskinfo-Struktur ist folgendermaßen aufgebaut:

```
TYPE DISKINFO_TR
STRUCT
    blocksize_w     : WORD; (* Größe eines logischen Blocks in Bytes *)
    blockanz_w      : WORD; (* Formatierte Gesamtkapazität in Blöcken *)
    blockused_w     : WORD; (* Belegte Kapazität in Blöcken *)
    blockfree_w     : WORD; (* Freie Kapazität in Blöcken *)
END_STRUCT
END_TYPE
```

Parameter

info_p:	Zeiger auf die Diskinfo-Struktur.
device_pc:	Zeiger auf den Devicenamen 📖 381. Leerstring (" ") für das Default-Device.

8.8.3.13 Gerätetreiber

Die Steuerung unterstützt verschiedene Geräte, auf die schreibend (w) und je nach Gerät lesend (r) zugegriffen werden. Der Zugriff erfolgt über ein Dateisystem. Eine Datei muss vor dem Zugriff durch die Funktion SysOpenFile() geöffnet werden. Außer auf dem Gerät "Drucker" können auf einem Gerät mehrere Dateien gleichzeitig geöffnet werden. Insgesamt sind vier gleichzeitig geöffnete Dateien möglich. Die Dateinamen müssen der DOS 8.3-Konvention entsprechen, also maximal 8 Zeichen für den Dateinamen und 3 Zeichen für die Datei-Erweiterung enthalten (Sonderzeichen dürfen nicht verwendet werden).

Gerät	Kennung	Zugriff	Beschreibung
Silicon-Disk	SD	rw	Dieser Treiber ermöglicht den Zugriff auf das Flash-PROM der Steuerung.
RAM-Disk	RD	rw	Dieser Treiber ermöglicht den Zugriff auf die so genannte RAM-Disk (batteriegepuffertes RAM).
Net-Disk	ND	rw	Dieser Treiber ermöglicht den Zugriff auf die Festplatte eines PC's. Das Gerät kann nur genutzt werden, wenn die Steuerung über eine Ethernet-Schnittstelle verfügt. Auf dem PC muss der Lenze ETC Data Server gestartet sein. Die SPS kann auf das Verzeichnis zugreifen, das in der Konfigurationsdatei des Servers eingetragen wurde. Um die Verbindung zwischen Steuerung und PC herzustellen, wird der Steuerung über die Funktion SetCurrentPath() die IP-Adresse des PC's bekannt gegeben. Dazu ist es notwendig, dass in den Netzwerkeinstellungen des PC's eine feste IP-Adresse eingetragen wird.
PC-Disk	PC	rw	Dieser Treiber ermöglicht den Zugriff auf die Festplatte eines PC's. Das Gerät kann nur verwendet werden, wenn auf dem PC eine Applikation läuft, die das Lenze Gateway benutzt (z. B. ETC-MMI). Mit Hilfe der Gateway Konfigurationssoftware kann ein Verzeichnis festgelegt werden, auf das die SPS dann zugreifen kann.
Programmspeicher (nur ETCxC)	PS	w	Dieser Treiber ermöglicht es DIN-Programme durch die SPS in den internen DIN-Programmspeicher zu schreiben. Hierbei ist zu beachten, dass der Dateiname gleichzeitig der Programmnummer entspricht und damit im DIN-Programm keine Programmnummer (mit %-Anweisung) enthalten sein darf. Beispiel: SysOpenFile('PS:12.din', O_WRONLY); öffnet ein DIN-Prog. mit der Programmnummer %12.

8.8.3.14 Globale Konstanten zu File IO-Funktionen

Konstante	Wert	Bedeutung
DSK_EEPROM_NOT_READY_KI	-1	Flash-PROM defekt
DSK_EEPROM_CHECKSUM_KI	-2	Checksummenfehler in der Flash-Disk
DSK_EEPROM_DISK_FULL_KI	-3	Flash-Disk ist voll, Datei kann nicht übernommen werden
DSK_ACCESS_DENIED_KI	-4	Datei ist bereits geöffnet, Zugriff verweigert
DSK_TOO_MANY_FILES_KI	-5	Zu viele gleichzeitig geöffnete Dateien, Zugriff verweigert
DSK_EEPROM_DIR_FULL_KI	-6	Flash-Disk-Inhaltsverzeichnis ist voll, zu viele Dateien
DSK_FILE_NOT_EXIST_KI	-7	Datei nicht vorhanden

Konstante	Wert	Bedeutung
DSK_INVALID_ACCESSMODE_KI	-8	Nicht erlaubter Zugriffsmodus bei SysOpenFile
DSK_UNKNOWN_EEPROM_TYPE_KI	-9	Unbekannten Flash-Typ gefunden
DSK_FILE_NOT_OPEN_KI	-10	Datei nicht zum Lesen oder Schreiben geöffnet
DSK_EEPROM_OUT_OF_MEMORY	-11	Flash-Disk: Nicht genug Speicher zum Anlegen der Flash-Disk
DSK_SRAM_TIMEOUT_KI	-91	RAM-Disk: XT3-Erweiterungskarte meldet sich nicht
DSK_SRAM_COM_ERROR_KI	-92	RAM-Disk: Kommunikationsfehler mit XT3-Erweiterungskarte
DSK_SRAM_CHECKSUM_KI	-93	Checksummenfehler in der RAM-Disk
DSK_SRAM_OUTOFMEMORY_KI	-94	RAM-Disk: Nicht genug Heap-Speicher zum Anlegen der File-Buffer
DSK_SRAM_DIR_FULL_KI	-95	Kein Platz mehr im RAM-Disk-Directory, zu viele Dateien
DSK_SRAM_DISK_FULL_KI	-96	RAM-Disk ist voll
DSK_SRAM_VALIDATE_ERR_KI	-97	Validieren der RAM-Disk: Datei war nicht zu retten und wurde gelöscht
DSK_SRAM_BATT_EMPTY_KI	-98	RAM-Disk: Batterie ist leer, bitte wechseln
DSK_SRAM_RAM_ERROR_KI	-99	RAM-Disk: RAM-Fehler im CMOS-RAM
DSK_UNKNOWN_DEVICE_KI	-110	Unbekannter Devicename
DSK_FLOPPY_V24_INUSE_KI	-111	V24-Schnittstelle zum Diskettenlaufwerk ist bereits anderweitig belegt
DSK_FLOPPY_INUSE_KI	-112	Diskettenlaufwerk ist nicht frei, Zugriff verweigert
DSK_FLOPPY_TIMEOUT_KI	-113	Timeout beim Zugriff auf das Diskettenlaufwerk
DSK_FLOPPY_PROG_ERROR_KI	-114	Fehler beim Zugriff auf das Diskettenlaufwerk
DSK_FLOPPY_TRANSMIT_ERROR_KI	-115	Diskettenlaufwerk meldet Übertragungsfehler
DSK_FLOPPY_OVERRUN_ERROR_KI	-116	Diskettenlaufwerk meldet V24-Overrun
DSK_FLOPPY_WRITE_PROTECT_KI	-117	Diskette ist schreibgeschützt
DSK_FLOPPY_NO_DISK_KI	-118	Keine Diskette eingelegt
DSK_FLOPPY_NOT_FORMATTED_KI	-119	Schreib-/Lesefehler auf der Diskette
DSK_FUNCTION_NOT_SUPPORTED_KI	-120	Die auszuführende Funktion wird vom Device-Treiber nicht unterstützt
DSK_INVALID_FILENAME_KI	-121	Dateiname enthält unzulässige Buchstaben
DSK_PRINTER_v24_INUSE_KI	-122	V24-Schnittstelle zum Drucker ist bereits anderweitig belegt
DSK_PRINTER_INUSE_KI	-123	Drucker wird bereits benutzt, Zugriff verweigert
DSK_PRINTER_TIMEOUT_KI	-124	Drucker nicht angeschlossen oder nicht bereit
DSK_BLOCK_OUT_OF_RANGE	-200	Datei-Blocknummer nicht im erlaubten Bereich
DSK_DUPLICATE_BLOCK_KI	-201	Datei-Block doppelt belegt
DSK_PCDISK_IOERR	-301	PC-Disk: Unbekannter IO-Fehler
DSK_DLL_TIMEOUT	-302	PC-Disk: Timeout bei IO-Operation
DSK_DLL_OUT_OF_MEMORY	-303	PC-Disk: Speichermangel bei IO-Operation
DSK_NET_NOT_CONFIGURED	-331	Net-Disk: Laufwerk nicht initialisiert
DSK_NET_COM_ERR	-332	Net-Disk: Kommunikationsfehler, falscher Befehlscode
DSK_NET_ANSWER_ERR	-333	Net-Disk: Kommunikationsfehler, falsche Quittung
DSK_NET_TIMEOUT_ERR	-334	Net-Disk: Timeout beim Zugriff auf das Laufwerk

Konstante	Wert	Bedeutung
DSK_NET_CANNOT_OPFEN_FILE	-335	Net-Disk: Server meldet Fehler bei FileOpen
DSK_NET_CANNOT_CLOSE_FILE	-336	Net-Disk: Server meldet Fehler bei FileClose
DSK_NET_HANDLE_ERROR	-337	Net-Disk: Server meldet Handle-Fehler
DSK_NET_READ_ERROR	-338	Net-Disk: Server meldet Fehler bei FileRead
DSK_NET_WRITE_ERROR	-339	Net-Disk: Server meldet Fehler bei FileWrite
DSK_NET_FILE_NOT_EXISTS	-340	Net-Disk: Datei nicht vorhanden
DSK_NET_DIR_NOT_CREATED	-341	Net-Disk: Server meldet Fehler bei CreateDir
DSK_NET_DIR_NOT_DELETED	-342	Net-Disk: Server meldet Fehler bei DeleteDir
DSK_NET_FILE_NOT_CREATED	-343	Net-Disk: Server meldet Fehler bei CreateFile
DSK_NET_FILE_NOT_DELETED	-344	Net-Disk: Server meldet Fehler bei DeleteFile
DSK_NET_BUSY	-345	Net-Disk: Laufwerk ist nicht frei, Zugriff verweigert
DSK_CMS_INIT_KI	-401	CMOS-RAM ungültig: Remanente Daten wurden gelöscht
DSK_CMS_BATT_EMPTY_KI	-402	CMOS-RAM: Batterie ist leer, bitte wechseln
DSK_CMS_OLD_BOOTLOADER_KI	-403	CMOS-RAM: alte Version des Bootloaders, bitte updaten
O_RDONLY	0	
O_WRONLY	1	

8.8.4 Speicherzugriffsfunktionen

8.8.4.1 DEFDATATYPES

Deklaration

```
FUNCTION DEFDATATYPES: BYTE
VAR_INPUT
    iBereich        : INT;
    sDescriptor     : STRING(255);
END_VAR
```

Beschreibung

Die Funktion definiert die Datentypen im DB2 und damit das notwendige Byte-Swapping beim Datenaustausch zwischen NCR und MMI.

Zurzeit werden zwei Bereiche unterschieden:

Bereich 1 von Wort 0 ... 127 enthält Daten von SPS an MMI, Bereich 2 von Wort 128 ... 191 enthält Daten von MMI an SPS. Zur Definition der Datentypen wird ein Descriptor-String verwendet. Der Descriptorstring hat folgendes Format:

<Anzahl><Typ><Anzahl><Typ>...

Anzahl ist eine Dezimalzahl und gibt die Menge der Daten vom nachfolgenden Typ an.

Typ ist ein Kleinbuchstabe, der die Größe des zugehörigen Datentyps wie folgt beschreibt:

Typ	Codesys-Datentyp	Größe
b	BYTE	8 Bit
w	BOOL, WORD, INT	16 Bit
d	DWORD, DINT, REAL	32 Bit
l	LREAL	64 Bit

Die Funktion liefert den Wert 1, wenn die Funktion erfolgreich ausgeführt wurde, ansonsten den Wert 0.

Beispiel

Am Anfang vom DB2 im Bereich 1 von *%MW2.0* bis *%MW2.16* liegen 10 BYTE, 2 WORD, 3 DINT und 1 LREAL. Im Bereich 2 von *%MW2.128* bis *%MW2.158* liegen 5 WORD, 8 DWORD und 5 WORD in dieser Reihenfolge. Die Definition erfolgt folgendermaßen:

```
DEFDATATYPES(1,'10b2w3d1l')
DEFDATATYPES(2,'5w8d10w')
```

Wenn eine Nachrichtenschnittstelle im DB2 zur Kommunikation zwischen SPS und HMI verwendet wird und in den Nachrichten andere Datentypen als WORD verwendet werden, sind weitergehende Anpassungen notwendig. Die ETC kennt zu diesem Zweck typengerechte Funktionen:

GET_BYTE	PUT_BYTE
GET_WORD	PUT_WORD
GET_INT	PUT_INT
GET_DWORD	PUT_DWORD
GET_DINT	PUT_DINT
GET_REAL	PUT_REAL
GET_LREAL	PUT_LREAL

Die Funktionen GET_... lesen den entsprechenden Datentyp von der angegebenen Adresse und führt das notwendige Byte-Swapping für das HMI durch. Die Funktionen PUT_... schreiben den entsprechenden Datentyp geswapped an die angegebene Adresse.

Voraussetzung dafür, dass die Daten dann richtig beim HMI ankommen, wenn die obigen Funktionen verwendet werden, ist die Definition der Nutzdaten der Nachricht als Byte mit DEFDATATYPES. Für das Standard-HMI sieht das dann wie folgt aus:

```
DEFDATATYPES(1,'96w4w52b2w'); (* 96 WORD, 4 WORD, 52 BYTE, 2 WORD *)
DEFDATATYPES(2,'32w4w52b2w'); (* 32 WORD, 4 WORD, 52 BYTE, 2 WORD *)
```

Die Definition ergibt sich aus der Festlegung der Daten im DB2, die zwischen ETC-MMI und SPS ausgetauscht werden:

Datenwort	Name	Richtung
000.00 – 015.15	Zustände (256 Bits)	PLC → HMI
016.00 – 079.15	Anzeigen (64 Datenwort)	PLC → HMI
080.00 – 095.15	Hinweise/statische Fehler (256 Bits)	PLC → HMI
096.00 – 096.15	Nachrichtenpuffer: Quittungszähler	PLC → HMI
097.00 – 097.15	Nachrichtenpuffer: Zähler 1	PLC → HMI
098.00 – 125.15	Nachrichtenpuffer: Nutzdaten (28 Datenworte)	PLC → HMI
126.00 – 126.15	Nachrichtenpuffer: Zähler 2	PLC → HMI
127.00 – 127.15	Überwachung SPS läuft	PLC → HMI
128.00 – 143.15	Zustände (256 Bits)	PLC ← HMI
144.00 – 159.15	Tasten (256 Bits)	PLC ← HMI
160.00 – 160.15	Nachrichtenpuffer: Quittungszähler	PLC ← HMI
161.00 – 161.15	Nachrichtenpuffer: Zähler 1	PLC ← HMI
162.00 – 189.15	Nachrichtenpuffer: Nutzdaten (28 Datenwort)	PLC ← HMI
190.00 – 190.15	Nachrichtenpuffer: Zähler 2	PLC ← HMI
191.00 – 191.15	Überwachung ETC-MMI läuft	PLC ← HMI

8.8.4.2 GET_BYTE, GET_WORD, GET_DWORD, GET_INT, GET_DINT, GET_REAL, GET_LREAL

Deklaration

```
FUNCTION GET_TYPE : BYTE
VAR_INPUT
    pAddress        : DINT;          (* Speicheradresse *)
END_VAR
```

Beschreibung

Die Funktionen GET_TYPE lesen den entsprechenden Datentyp TYPE von der angegebenen Adresse und führen das notwendige Byte-Swapping durch (siehe auch DEFDATATYPES).

8.8.4.3 MEMCOMP

Deklaration

```
FUNCTION MEMCOMP: INT
VAR_INPUT
    pMem1           : DINT;          (* Speicheradresse *)
    pMem2           : DINT;          (* Speicheradresse *)
    diSize          : DINT;          (* Anzahl Bytes *)
END_VAR
```

Beschreibung

Es werden die Speicherbereiche Mem1 und Mem2 byteweise verglichen.

Returnwert

0: wenn der Inhalt gleich ist

1: wenn der Inhalt von Mem1 > Mem2 ist

-1: wenn der Inhalt von Mem1 < Mem2 ist

8.8.4.4 MEMCOPY

Deklaration

```
FUNCTION MEMCOPY: BOOL
VAR_INPUT
    ziel_p          : DINT;          (* Speicheradresse *)
    quelle_p        : DINT;          (* Speicheradresse *)
    anzahl_di       : DINT;          (* Anzahl der Bytes *)
END_VAR
```

Beschreibung

Es werden *anzahl_di* Bytes ab Speicheradresse *quelle_p* gelesen und ab Speicheradresse *ziel_p* eingetragen.

Der Returnwert der Funktion hat keine Bedeutung.

Beispiel

```
ret_bit         : BOOL;
feld1_ab        : ARRAY[0..10] OF BYTE;
feld2_ab        : ARRAY[0..5] OF BYTE;

MEMCOPY(ADR(feld1_ab[4]), ADR(feld2_ab[2]), 4);
MEMCOPY(ADR(feld2_ab[0]), ADR(feld1_ab[2]), SIZEOF(feld2_ab));
```

8.8.4.5 MEMSET

Deklaration

```
FUNCTION MEMSET: BOOL
VAR_INPUT
    pMem        : DINT;        (* Speicheradresse *)
    bValue      : DINT;        (* Wert *)
    dwSize      : DINT;        (* Anzahl der Bytes *)
END_VAR
```

Beschreibung

Es werden *dwSize* Bytes ab Speicheradresse *pMem* mit Wert *bValue* geschrieben.

Beispiel

```
feld_ab             : ARRAY[0..10] OF BYTE;
MEMSET( ADR(feld_ab[0]), 0, SIZEOF(feld_ab) );
```

8.8.4.6 OVESWAPPED

Deklaration

```
FUNCTION MOVESWAPPED: DINT
VAR_INPUT
    pDestination    : DINT;          (* Speicheradresse *)
    pSource         : DINT;          (* Speicheradresse *)
    sDescstring     : STRING(255);   (* Datenbeschreibungsstring *)
END_VAR
```

Beschreibung

Die Funktionen kopiert Daten von der Speicheradresse *pDestination* zu Speicheradresse *pSource*. Die Daten werden entsprechend des angegebenen Desciptor-Strings geswappt kopiert.

Der Descriptorstring hat folgendes Format:

<Anzahl><Typ><Anzahl><Typ>...

Anzahl ist eine Dezimalzahl und gibt die Menge der Daten vom nachfolgenden Typ an.

Typ ist ein Kleinbuchstabe, der die Größe des zugehörigen Datentyps wie folgt beschreibt:

Typ	Codesys-Datentyp	Größe
b	BYTE	8 Bit
w	BOOL, WORD, INT	16 Bit
d	DWORD, DINT, REAL	32 Bit
l	LREAL	64 Bit

Die Funktion liefert als Returnwert die Anzahl der kopierten Bytes.

Beispiel

```
TYPE DATA_TR :
STRUCT
    var1            : DINT;
    var2            : DINT;
END_STRUCT
END_TYPE

byte_di             : DINT;
descstring_s        : STRING(10) := '2d';
data_st             : DATA_TR := (var1:=16#FF, var2:=16#123456);
buf_ab              : ARRAY[0..20] OF BYTE;

byte_di := MOVESWAPPED(ADR(buf_ab), ADR(data_st), descstring_s);
```

Ergebnis:

```
byte_di = 8, buf_ab[0] = 16#FF, buf_ab[4] = 16#56, buf_ab[5] = 34, buf_ab[6] := 16#12
```

Alle anderen Elemente des Arrays sind gleich 0.

8.8.4.7 PUT_BYTE, PUT_WORD, PUT_DWORD, PUT_INT, PUT_DINT, PUT_REAL, PUT_LREAL

Deklaration

```
FUNCTION PUT_TYPE: BYTE
VAR_INPUT
    pAddress        : DINT;         * Speicheradresse *)
    Value           : TYPE;         (* zu schreibender Wert *)
END_VAR
```

Beschreibung

Die Funktionen PUT_TYPE schreiben den entsprechenden Datentyp TYPE geswappt an die angegebene Adresse (siehe auch DEFDATATYPES).

8.8.5 CANopen Funktionen

8.8.5.1 CopDefineDS403

Deklaration

```
FUNCTION CopDefineDS403: BOOL
VAR_INPUT          (* Def. die Global-Database für ein CanOpen-Bedienteil nach DS403 *)
    NodeID         : BYTE;
    DataBase       : DINT;
    DataSize       : WORD;
    Datatypes      : STRING(255);
    ReadDataObject : WORD;
    WriteDataObject: WORD;
END_VAR
```

Parameter

NodeID	Knotennummer des Bedienteils
DataBase	Adresse der Global-Database mit den Ein-/Ausgabedaten des Bedienteils
DataSize	Größe der Global-Database in Byte, max. 1020 Byte
Datatypes	Descriptor-String zur Definition der Datentypen in der Global-Database
ReadDataObject	Objektnummer zum Lesen der ”Local-Database”
WriteDataObject	Objektnummer zum Beschreiben der ”Local-Database”

Beschreibung

Diese Funktion gibt dem Laufzeitsystem einen Datenbereich der SPS zum Datenaustausch mit einem CANopen-Bedienteil bekannt. Dieser Datenbereich wird in der SPS am einfachsten als Datenstruktur oder als Array angelegt und mit den anzuzeigenden Daten initialisiert. Nach Aufruf der Funktion wird diese Global-Database einmal komplett in die Local-Database des entsprechenden Knotens übertragen. Anschließend werden nur noch die Änderungen der Anzeigedaten von der Steuerung übertragen.

ReadDataObject und WriteDataObject sind die Objektnummern des Objektverzeichnisses des CANopen-Displays, die vom Laufzeitsystem zum Lesen und Schreiben der Local-Database verwendet werden müssen.

Der Descriptorstring hat folgendes Format:

<Anzahl><Typ><Anzahl><Typ>...

Anzahl ist eine Dezimalzahl und gibt die Menge der Daten vom nachfolgenden Typ an.

Typ ist ein Kleinbuchstabe, der die Größe des zugehörigen Datentyps wie folgt beschreibt:

Typ	Codesys-Datentyp	Größe
b	BYTE	8 Bit
w	BOOL, WORD, INT	16 Bit
d	DWORD, DINT, REAL	32 Bit

Der Returnwert ist TRUE, wenn die Funktion erfolgreich ausgeführt werden konnte. Andernfalls konnte die Funktion, z. B. wegen Speichermangel, nicht ausgeführt werden.

Hinweis!

Die Database sollte nur so groß sein, wie die tatsächlich verwendeten Daten es erfordern.

Die SPS sollte immer nur die Daten in der Global-Database aktualisieren, die auch aktuell im Display zur Anzeige kommen, um die Buslast auf dem CAN-Bus nicht unnötig zu erhöhen. Zyklische Änderungen in jedem SPS-Takt sollten generell vermieden werden.

Beispiel

```
TYPE DATABASE_TR :
STRUCT
    dummy_dw        : DWORD;
    pad1_dw         : DWORD;
    pad2_b          : BYTE;
    pad3_b          : BYTE;
    pad4_w          : WORD;
END_STRUCT
END_TYPE
data_st             : DATABASE_TR;
COPDEFINEDS403( 127, ADR(data_st), SIZEOF(data_st), '2d2b1w', 16#6700, 16#8100 );
```

Die Funktion definiert für ein CANopen-Bedienteil mit der Knotennummer 127, das entsprechend in der Steuerungskonfiguration von CoDeSys eingefügt wurde, eine Global-Database mit den durch den Descriptor-String definierten Elementen.

8.8.5.2 CopXDefineDS403 (nur ETCxM)

Deklaration

```
FUNCTION CopXDefineDS403: BOOL          (* Definiert die Global-Database für ein CANo-
                                           pen-Bedienteil nach DS403 für 1/2 CAN-Bus *)
VAR_INPUT
    CanNum_b        : BYTE
    NodeID_b        : BYTE
    DataBase_p      : DINT;
    Len_w           : WORD;
    Datatypes_s     : STRING(255);
    ReadDataObject  : WORD;
    WriteDataObject : WORD;
END_VAR
```

Parameter

CanNum_b:	Nummer des CAN-Bus (1 oder 2)
NodeID_b:	Knotennummer des Bedienteils
DataBase_p:	Adresse der Global-Database mit den Ein-/Ausgabedaten des Bedienteils
Len_w:	Größe der Global-Database in Byte, max. 1020 Byte
Datatypes_s:	Descriptor-String zur Definition der Datentypen in der Global-Database
ReadDataObject:	Objektnummer zum Lesen der ”Local-Database”
WriteDataObject:	Objektnummer zum Beschreiben der ”Local-Database”

Beschreibung

Die Funktionalität dieser Funktion entspricht der Funktion CopDefineDS403(). Sie enthält einen zusätzlichen Parameter *CanNum_b*, der die Nummer des betroffenen CAN-Bus bestimmt.

8.8.5.3 CopGetNodeState (nur ETCxM)

Deklaration

```
FUNCTION CopGetNodeState: BOOL
VAR_INPUT
    NodeID_b        : BYTE;             * CANopen-ID des Moduls *)
END_VAR
```

Parameter

NodeID_b:	Knotennummer des CANopen-Knotens

Beschreibung

Der Returnwert TRUE signalisiert, dass das entsprechende CANopen-Modul am 1. CAN-Bus vorhanden und im Zustand ”Operational” ist. Ansonsten wird FALSE zurückgeliefert. Diese Erkennung funktioniert nur, wenn das Nodeguarding für das entsprechende CANopen-Gerät aktiviert ist .

8.8.5.4 CopReadObjekt (ETCxC und ETCxM an CAN1), DrvReadObject (ETCxC an CAN2)

Deklaration

```
FUNCTION CopReadObject: BOOL            (ETCxC und ETCxM auf CAN1)
(* Auslesen von Objekten des Objektverzeichnisses von CANopen-Geräten an CAN1 *)
VAR_INPUT
    NodeID      : BYTE;
    ObjectNum   : WORD;
    SubIndex    : BYTE;
    DataType    : WORD;
    Buffer      : DINT;
    BufSize     : WORD;
    Status      : POINTER TO BYTE;
END_VAR
```

```
FUNCTION DrvReadObject: BOOL            (ETCxC auf CAN2)
(* Auslesen von Objekten des Objektverzeichnisses von CANopen-Geräten an CAN2 *)
VAR_INPUT
    NodeID      : BYTE;
    ObjectNum   : WORD;
    SubIndex    : BYTE;
    DataType    : WORD;
    Buffer      : DINT;
    BufSize     : WORD;
    Status      : POINTER TO BYTE;
END_VAR
```

Parameter

NodeID	Knotennummer des CANopen-Knotens
ObjectNum	Nummer des Objekts, welches gelesen werden soll
SubIndex	Index des Unterobjekts, welches gelesen werden soll
DataType	Datentyp nach CiA DS301
Buffer	Adresse des Puffers, in dem die gelesenen Daten abgelegt werden
BufSize	Größe des Puffers in Byte
Status	Adresse einer Variablen zur Ablage des Transferstatus

Beschreibung

Diese Funktion ermöglicht es der SPS, einzelne Objekte von CANopen-Geräten auszulesen. Der Lesevorgang erfolgt dabei im Hintergrund während die SPS weiter läuft. Die Transferqueue kann bis zu 15 Schreib-/Leseaufträge puffern.

Durch Abfrage des Transferstatus kann die SPS überprüfen, ob die Übertragung noch läuft, beendet ist, oder ein Fehler bei der Übertragung aufgetreten ist. Sollen mehrere Transferaufträge parallel ausgeführt werden, so muss für jeden Auftrag eine eigene Statusvariable definiert werden.

Der Returnwert ist TRUE, wenn der Auftrag zum Auslesen in die Transferqueue übertragen werden konnte, andernfalls ist die Transferqueue bereits voll.

Status	Bedeutung
0	Inaktiv
1	Auftrag in der Transferqueue
2	Transfer aktiv
3	Transfer erfolgreich beendet
4	Transfer abgebrochen
5	Timeout

Beispiel

```
status_b
devicetype_dw

CopReadObject(1, 16#1000, 0, COP_UNSIGNED32_KW, ADR(devicetype_dw),
SIZEOF(devicetype_dw), ADR(status_b));
```

8.8.5.5 CopXReadObject (ETCxM an CAN2)

Deklaration

```
FUNCTION CopXReadObject: BOOL          (* Auslesen von Objekten des Objektverzeichnisses von CANopen-Geräten an CAN2 *)

VAR_INPUT
    CanNum_b      : BYTE;
    NodeID_b      : BYTE;
    ObjectNum_w   : WORD;
    SubIndex_b    : BYTE;
    DataType_w    : WORD;
    Buffer_p      : DINT;
    BufSize_w     : WORD;
    Status_pb     : POINTER TO BYTE;
END_VAR
```

Parameter

CanNum_b:	Nummer des CAN-Bus (1 oder 2)
NodeID_b:	Knotennummer des CANopen-Knotens
ObjectNum_w:	Nummer des Objekts, das gelesen werden soll
SubIndex_b:	Index des Unterobjekts, das gelesen werden soll
DataType_w:	Datentyp nach CiA DS301
Buffer_p:	Adresse des Puffers, in dem die gelesenen Daten abgelegt werden
BufSize_w:	Größe des Puffers in Byte
Status_pb:	Adresse einer Variablen zur Ablage des Transferstatus

Beschreibung

Die Funktionalität dieser Funktion entspricht voll der Funktion CopReadObjekt(). Sie enthält einen zusätzlichen Parameter *CanNum_b*, der die Nummer des betroffenen CAN-Bus bestimmt.

8.8.5.6 CopWriteObject (ETCxC und ETCxM auf CAN1), DrvWriteObject (ETCxC auf CAN2)

Deklaration

```
FUNCTION CopWriteObject: BOOL          (ETCxC und ETCxM auf CAN1)
(* Beschreiben von Objekten im Objektverzeichnis von CANopen-Geräten an CAN1 *)
VAR_INPUT
    NodeID      : BYTE;
    ObjectNum   : WORD;
    SubIndex    : BYTE;
    DataType    : WORD;
    Buffer      : DINT;
    BufSize     : WORD;
    Status      : POINTER TO BYTE;
END_VAR
```

```
FUNCTION DrvWriteObject: BOOL          (ETCxC auf CAN2)
(* Beschreiben von Objekten im Objektverzeichnis von CANopen-Geräten an CAN2 *)
VAR_INPUT
    NodeID      : BYTE;
    ObjectNum   : WORD;
    SubIndex    : BYTE;
    DataType    : WORD;
    Buffer      : DINT;
    BufSize     : WORD;
    Status      : POINTER TO BYTE;
END_VAR
```

Parameter

NodeID	Knotennummer des CANopen-Gerätes
ObjectNum	Nummer des Objekts, welches geschrieben werden soll
SubIndex	Index des Unterobjekts, welches geschrieben werden soll
DataType	Datentyp nach CiA DS301
Buffer	Adresse des Puffers, in dem die zu schreibenden Daten abgelegt sind
BufSize	Anzahl zu schreibender Bytes
Status	Adresse einer Variablen zur Ablage des Transferstatus

Beschreibung

Diese Funktion ermöglicht es der SPS, einzelne Objekte von CANopen-Geräten zu beschreiben. Der Schreibvorgang erfolgt dabei im Hintergrund während die SPS weiter läuft. Die Transferqueue kann bis zu 15 Schreib-/Leseaufträge puffern.

Durch Abfrage des Transferstatus kann die SPS überprüfen, ob die Übertragung noch läuft, beendet ist oder ein Fehler bei der Übertragung aufgetreten ist. Sollen mehrere Transferaufträge parallel ausgeführt werden, so muss für jeden Auftrag eine eigene Statusvariable definiert werden.

Die Funktion liefert TRUE, wenn der Auftrag zum Schreiben in die Transferqueue übertragen wurde, andernfalls ist die Transferqueue bereits voll.

Status	Bedeutung
0	Inaktiv
1	Auftrag in der Transferqueue
2	Transfer aktiv
3	Transfer erfolgreich beendet
4	Transfer abgebrochen
5	Timeout

Beispiel

```
status_b        : BYTE;
wert_di         : DINT := 1000;

CopWriteObject(5, 16#607A, 0, COP_INTEGER32_KW, ADR(wert_di), SIZEOF(wert_di),
ADR(status_b));
```

Eintragen der Zielposition für einen DS402-Antrieb.

8.8.5.7 CopXWriteObject (ETCxM auf CAN2)

Deklaration

```
FUNCTION CopXWriteObject: BOOL          (* Beschreiben von Objekten im Objektverzeichnis von CANopen-Geräten an CAN2 *)

VAR_INPUT
    CanNum_b        : BYTE;
    NodeID_b        : BYTE;
    ObjectNum_w     : WORD;
    SubIndex_b      : BYTE;
    DataType_w      : WORD;
    Buffer_p        : DINT;
    BufSize_w       : WORD;
    Status_pb       : POINTER TO BYTE;
END_VAR
```

Parameter

CanNum_b:	Nummer des CAN-Bus (1 oder 2)
NodeID_b:	Knotennummer des CANopen-Gerätes
ObjectNum_w:	Nummer des Objekts, das geschrieben werden soll
SubIndex_b:	Index des Unterobjekts, das geschrieben werden soll
DataType_w:	Datentyp nach CiA DS301
Buffer_p:	Adresse des Puffers, in dem die zu schreibenden Daten abgelegt sind
BufSize_w:	Anzahl zu schreibender Bytes
Status_pb:	Adresse einer Variablen zur Ablage des Transferstatus

Beschreibung

Die Funktionalität dieser Funktion entspricht der Funktion CopWriteObject. Sie enthält einen zusätzlichen Parameter *CanNum_b*, der die Nummer des betroffenen CAN-Bus bestimmt.

8.8.5.8 CopXSetOperational (nur ETCxM)

Deklaration

```
FUNCTION CopXSetOperational: BOOL        (* Setzen des CANopen-Teilnehmers in Zustand Operational *)
VAR_INPUT
    CanNum_b    : BYTE;
    NodeID_b    : BYTE;
    Status_pb   : POINTER TO BYTE;
END_VAR
```

Parameter

CanNum_b:	Nummer des CAN-Bus (1 oder 2)
NodeID_b:	Knotennummer des CANopen-Gerätes
Status_pb:	Adresse einer Variablen zur Ablage des Transferstatus

Beschreibung

Die Funktion ermöglicht es der SPS, einzelne CANopen-Teilnehmer in den Zustand Operational zu setzen. Voraussetzung dafür ist, dass der Teilnehmer in der CoDeSys-Steuerungskonfiguration eingetragen wurde. Das Versenden der Telegramme erfolgt dabei im Hintergrund, während die SPS weiterläuft.

Durch Abfrage des Transferstatus kann die SPS überprüfen, ob die Nachricht gesendet wurde oder ein Fehler aufgetreten ist. Sollen mehrere Transferaufträge parallel ausgeführt werden, so muss für jeden Auftrag eine eigene Statusvariable definiert werden.

Die Funktion liefert TRUE, wenn der Auftrag zum Schreiben in die Transferqueue übertragen wurde. Anderenfalls ist die Transferqueue bereits voll.

Status	Bedeutung
0	Inaktiv
1	Auftrag in der Transferqueue
2	Transfer aktiv
3	Transfer erfolgreich beendet
4	Transfer abgebrochen

Beispiel

```
status_b        BYTE;
wert_di         : DINT := 1000;

CopXSetOperational(1,5, ADR(status_b));
```

Der Knoten mit der Adresse 5 am 1. CAN-Bus wird in den Zustand Operational gesetzt.

8.8.5.9 CopXSetPreoperational (nur ETCxM)

Deklaration

```
FUNCTION CopXSetPreoparational: BOOL     (* Setzen des CANopen-Teilnehmers in Zustand Preoperational *)
VAR_INPUT
    CanNum_b    : BYTE;
    NodeID_b    : BYTE;
    Status_pb   : POINTER TO BYTE;
END_VAR
```

Parameter

CanNum_b:	Nummer des CAN-Bus (1 oder 2)
NodeID_b:	Knotennummer des CANopen-Gerätes
Status_pb:	Adresse einer Variablen zur Ablage des Transferstatus

Beschreibung

Diese Funktion ermöglicht es der SPS, einzelne CANopen-Teilnehmer in den Zustand Preoperational zu setzen. Voraussetzung hierfür ist, dass der Teilnehmer in der CoDeSys-Steuerungskonfiguration eingetragen wurde. Das Versenden des Telegramms erfolgt dabei im Hintergrund, während die SPS weiterläuft.

Durch Abfrage des Transferstatus kann die SPS überprüfen, ob die Nachricht gesendet wurde oder ein Fehler aufgetreten ist. Sollen mehrere Transferaufträge parallel ausgeführt werden, so muss für jeden Auftrag eine eigene Statusvariable definiert werden.

Die Funktion liefert TRUE, wenn der Auftrag zum Schreiben in die Transferqueue übertragen wurde. Anderenfalls ist die Transferqueue bereits voll.

Status	Bedeutung
0	Inaktiv
1	Auftrag in der Transferqueue
2	Transfer aktiv
3	Transfer erfolgreich beendet
4	Transfer abgebrochen

Beispiel

```
status_b    : BYTE;
wert_di     : DINT := 1000;

CopXSetPreoperational(1,5, ADR(status_b));
```

Der Knoten mit der Adresse 5 am 1. CAN-Bus wird in den Zustand Preoperational gesetzt.

8.8.5.10 CopXEnableSync (nur ETCxM)

Deklaration

```
FUNCTION CopXEnableSync: DINT     (* Freigabe der Sync-Nachricht *)
VAR_INPUT
    CanNum_uc   : BYTE;
END_VAR
```

Parameter

CanNum_uc	Nummer des CAN-Bus (1 oder 2)

Beschreibung

Mit Aufruf dieser Funktion wird das Versenden der CANopen-Sync-Nachricht durch die SPS aktiviert. Die Sync-Nachrichten werden mit der Zykluszeit der aufrufenden PLC-Task versendet.

Die Funktion liefert folgende Werte zurück:

Returnwert	Bedeutung
0	Funktion erfolgreich ausgeführt
-1	CAN-Task ist nicht bereit
-3	Fehlerhafte CAN-Nummer
-4	Die Funktion ist schon frei gegeben
-5	Sendepuffer des CAN-Controllers ist schon belegt

8.8.5.11 CopXDisableSync (nur ETCxM)

Deklaration

```
FUNCTION CopXDisableSync : DINT          (* Sperren des Versenden der Sync-Nachrichten *)
VAR_INPUT
    CanNum_uc       : BYTE;
END_VAR
```

Parameter

CanNum_uc	Nummer des CAN-Bus (1 oder 2)

Beschreibung

Mit Aufruf dieser Funktion wird das Versenden der CANopen-Sync-Nachricht durch die SPS deaktiviert. Das Sperren der Sync-Nachrichten kann nur von der PLC-Task erfolgen, die das Versenden aktiviert hat.

Die Funktion liefert folgende Werte zurück:

Returnwert	Bedeutung
0	Funktion erfolgreich ausgeführt
-1	CAN-Task ist nicht bereit
-3	Fehlerhafte CAN-Nummer
-4	Die Funktion ist nicht aktiv
-5	Das Versenden der Sync-Nachrichten wurde von einer anderen Task frei gegeben.
-6	Der Sendepuffer des CAN-Controllers ist mit einer anderen COB-ID belegt.

8.8.5.12 CopyChannelDisplayData (nur ETCxC)

Deklaration

```
FUNCTION CopyChannelDisplayData: BOOL    (* aktualisiert die kanalabhängigen Daten des NCR's in der übergebenen Struktur *)
VAR_INPUT
    kanal_b         : BYTE;
    data_pst        : POINTER TO CHANNELDATA_TR;
END_VAR
```

Parameter

kanal_b	Auswahl des NC-Kanals (0/1)
data_pst	Zeiger auf eine Variable vom Typ CHANNELDATA_TR

Beschreibung

Die Funktion aktualisiert die Werte der in der Struktur enthaltenen Variablen.

8.8.5.13 CopyCyclicDisplayData (nur ETCxC)

Deklaration

```
FUNCTION CopyCyclicDisplayData: BOOL    (* aktualisiert die Daten in der übergebenen Struktur *)
VAR_INPUT
    data_pst    : POINTER TO CYCLICDATA_TR;
END_VAR
```

Parameter

data_pst	Zeiger auf eine Variable vom Typ CYCLICDATA_TR

Beschreibung

Die Funktion aktualisiert die Werte der in der Struktur enthaltenen Variablen.

8.8.5.14 CopyDiagDisplayData (nur ETCxC)

Deklaration

```
FUNCTION CopyDiagDisplayData: BOOL    (* aktualisiert die Daten der Struktur *)
VAR_INPUT
    data_pst    : POINTER TO DIAGDATA_TR;
END_VAR
```

Parameter

data_pst	Zeiger auf eine Variable vom Typ CYCLICDATA_TR

Beschreibung

Die Funktion aktualisiert die Werte der in der Struktur enthaltenen Variablen.

8.8.5.15 Globale Konstanten zu CANopen Funktionen

Diese Konstanten sind als Übergabeparameter "DataType" für die Funktionen CopReadObject() und CopWriteObject() gedacht. Die Werte der Konstanten entsprechen CiA DS301 "Application Layer and Communication Profile".

Konstante	Wert	Bedeutung
COP_BOOLEAN_KW	16#1	
COP_INTEGER8_KW	16#2	
COP_INTEGER16_KW	16#3	
COP_INTEGER32_KW	16#4	
COP_UNSIGNED8_KW	16#5	
COP_UNSIGNED16_KW	16#6	
COP_UNSIGNED32_KW	16#7	
COP_REAL32_KW	16#8	
COP_VISIBLE_STRING_KW	16#9	
COP_OCTET_STRING_KW	16#A	
COP_UNICODE_STRING_KW	16#B	
COP_BIT_STRING_KW	16#E	
COP_REAL64_KW	16#11	
Definitionen für Systemvariable db0_can_errorstatus_adw		
CAN_TXOVERRUN_KDW	16#20000	Sendequeue ist voll
CAN_OVERRUN_KDW	16#10000	Overrun beim Empfang
CAN_BITERR_KDW	16#C000	Transmit-Bit-Error
CAN_ACKERR_KDW	16#2000	Acknowledge-Error
CAN_CRCERR_KDW	16#1000	CRC-Error
CAN_FORMERR_KDW	16#0800	Message-Format-Error
CAN_STUFERR_KDW	16#0200	Transmit-Error-Counter >= 96
CAN_RXWARN_KDW	16#0100	Recieve-Error-Counter >= 96
CAN_IDLE_KDW	16#0040	kein Fehler
CAN_FCS_KDW	16#0030	Fault-Confinement-State
CAN_BOFFINT_KDW	16#0004	Bus-Off
CAN_ERRINT_KDW	16#0002	Error-Interrupt
CAN_WAKEINT_KDW	16#0001	kein Fehler

8.8.6 CAN-Funktionen (nur ETCxM)

8.8.6.1 DefineCanMsg (nur ETCxM)

Deklaration

```
FUNCTION DefineCanMsg: DINT
VAR_INPUT
    CanNum_w      : WORD;     (* Nummer der CAN-Schnittstelle - 1 oder 2 *)
    WriteCobId_w  : WORD;     (* CobId: ETCxM → CAN-Teilnehmer *)
    ReadCobId_w   : WORD;     (* CobId: CAN-Teilnehmer → ETCxM *)
    QueueSize_w   : WORD;     (* Anzahl Nachrichten, die gepuffert werden *)
END_VAR
```

Beschreibung

Anlegen und Initialisieren einer Struktur für die Verwaltung von VAN-Transfers. Damit ist es möglich, beliebige CAN-Nachrichten zu empfangen oder zu versenden. Beim Aufruf der Funktion wird überprüft, ob die angegebenen CobId noch nicht in Benutzung sind. Dabei kann die CobId auch in der Steuerungskonfiguration von CoDeSys bei einem CANopen-Modul eingetragen sein. ReadCobId_w bzw. WriteCobId_w gleich 0 bedeutet, dass vorerst keine CobId benötigt wird.

Die Funktion liefert ein Handle auf die Verwaltungsstruktur zurück. Wird 0 zurückgeliefert, konnte keine Verwaltungsstruktur mit den angegebenen Parametern angelegt werden.

Beispiel

```
handle_pr          : DINT;

handle_pr = DefineCanMsg(1, 1014, 1114, 32);
```

8.8.6.2 AddCobIdCanMsg (nur ETCxM)

Deklaration

```
FUNCTION AddCobIdCanMsg: BOOL
VAR_INPUT
    handle_pr      : DINT;      (* Handle auf die Verwaltungsstruktur *)
    CobId_w        : WORD;      (* CobId *)
    Read_bit       : BOOL;      (* Richtung, TRUE für CAN-Teilnehmer ETCxM *)
END_VAR
```

Beschreibung

Mit dieser Funktion kann eine CobId in die Verwaltungsstruktur (s. Funktion DefineCanMsg()) für CAN-Transfers hinzugefügt werden. Die Funktion liefert TRUE, wenn die CobId erfolgreich in die Verwaltungsstruktur eingetragen wurde. Wenn die CobId schon in Benutzung ist oder die maximale Anzahl (32) der möglichen CobId erreicht ist, wird FALSE zurückgeliefert.

Beispiel

```
handle_pr          : DINT;
FctReturn_bit      : BOOL;

handle_pr = DefineCanMsg(1, 1014, 1114, 32);
FctReturn_bit = AddCobIdCanMsg(handle_pr, 1024, 1);
```

8.8.6.3 DelCobIdCanMsg (nur ETCxM)

Deklaration

```
FUNCTION DelCobIdCanMsg: BOOL
VAR_INPUT
    handle_pr      : DINT;      (* Handle auf die Verwaltungsstruktur *)
    CobId_w        : WORD;      (* CobId *)
END_VAR
```

Beschreibung

Mit dieser Funktion wird eine CobId aus der Verwaltungsstruktur (s. Funktion DefineCanMsg()) für CAN-Transfers entfernt. Der Returnwert der Funktion ist ohne Bedeutung.

Beispiel

```
handle_pr           : DINT;

handle_pr = DefineCanMsg(1, 1014, 1114, 32);
DelCobIdCanMsg(handle_pr, 1014);
```

8.8.6.4 CloseCanMsg (nur ETCxM)

Deklaration

```
FUNCTION CloseCanMsg: BOOL
VAR_INPUT
    handle_pr       : DINT;          (* Handle auf die Verwaltungsstruktur *)
END_VAR
```

Beschreibung

Mit dieser Funktion wird die Verwaltungsstruktur für CAN-Transfers gelöscht (s. Funktion Define CanMsg()= und der Speicher wieder frei gegeben. Der Returnwert der Funktion ist ohne Bedeutung.

Beispiel

```
handle_pr

handle_pr = DefineCanMsg(1, 1014, 1114, 32);
CloseCanMsg(handle_pr);
```

8.8.6.5 ClearCanMsg (nur ETCxM)

Deklaration

```
FUNCTION ClearCanMsg: BOOL
VAR_INPUT
    handle_pr       : DINT;          (* Handle auf die Verwaltungsstruktur *)
END_VAR
FUNCTION
```

Beschreibung

Mit dieser Funktion kann das Empfangs-FIFO gelöscht werden (s. Funktion DefineCanMsg()). Der Returnwert der Funktion ist ohne Bedeutung.

Beispiel

```
handle_pr           : DINT;

handle_pr = DefineCanMsg(1, 1014, 1114, 32);
ClearCanMsg(handle_pr);
```

8.8.6.6 ReadCanMsg (nur ETCxM)

Deklaration

```
FUNCTION ReadCanMsg: INT
VAR_INPUT
    handle_pr       : DINT;     (* Handle auf die Verwaltungsstruktur *)
    dest_pr         : DINT;     (* Adresse auf die Datenstruktur *)
END_VAR
```

Beschreibung

Mit dieser Funktion kann eine CAN-Nachricht aus dem Empfangspuffer gelesen werden. Der Empfangspuffer ist als FIFO implementiert. Die Struktur für Empfangsdaten hat folgenden Aufbau:

```
TYPE CAN_MSG_TR
STRUCT
    cobid_w         : WORD;
    size_b          : BYTE;
    rtr_bit         : BOOL;
    data_ab         : ARRAY [0..7] OF BYTE;
END_STRUCT
END_TYPE
```

Der Variablen dest_pr muss immer die Adresse einer Variablen des Typs CAN_MSG_TR übergeben werden (s. Beispiel). Diese Funktion kehrt in jedem Fall sofort zurück. Die Bedeutung der Returnwerte ist der folgenden Tabelle zu entnehmen.

Returnwert	Bedeutung
-2	Ungültiges Handle
0	Keine neue Nachricht im FIFO
1	Neue Nachricht vorhanden, keine weitere Nachricht im FIFO
2	Neue Nachricht vorhanden und weitere Nachrichten im FIFO

Beispiel

```
handle_pr       : DINT;
received_bit    : BOOL;
msg_st          : CAN_MSG_TR;

handle_pr = DefineCanMsg(1, 1014, 1114, 32);
handle_pr = DefineCanMsg(1, 1014, 1114, 32);
```

8.8.6.7 WriteCanMsg (nur ETCxM)

Deklaration

```
FUNCTION WriteCanMsg: INT
VAR_INPUT
    handle_pr       : DINT;     (* Handle auf die Verwaltungsstruktur *)
    dest_pr         : DINT;     (* Adresse auf die Datenstruktur *)
END_VAR
```

Beschreibung

Mit dieser Funktion kann eine CAN-Nachricht in den Sendepuffer geschrieben werden. Der Sendepuffer ist als FIFO implementiert. Die Datenstruktur hat folgenden Aufbau:

```
TYPE CAN_MSG_TR
STRUCT
    cobid_w         : WORD;
    size_b          : BYTE;
    rtr_bit         : BOOL;
    data_ab         : ARRAY [0..7] OF BYTE;
END_STRUCT
END_TYPE
```

Der Variablen source_pr muss immer die Adresse einer Variablen des Typs CAN_MSG_TR übergeben werden (s. Beispiel). Diese Funktion kehrt in jedem Fall sofort zurück, auch wenn das FIFO voll ist. Die Bedeutung der Returnwerte ist der folgenden Tabelle zu entnehmen.

Returnwert	Bedeutung
-3	Nicht definierte CobId in Datenstruktur
-2	Ungültiges Handle oder kein Speicherplatz für Nachricht vorhanden
-1	FIFO voll
0	Nachricht im FIFO eingetragen

Beispiel

```
handle_pr       : DINT;
ret_bit         : BOOL;
src_st          : CAN_MSG_TR;

handle_pr = DefineCanMsg(1, 1014, 1114, 32);
src_st.cobid_w := 1014;
src_st.size_b := 1;
src_st.rtr_bit := FALSE;
src_st.data_ab[0] := 27;
ret_bit = WriteCanMsg(handle_pr, ADR(src_st));
```

8.8.7 MMI-Kommunikationsfunktionen

8.8.7.1 GetApplicationMessage

Deklaration

```
FUNCTION_BLOCK GetApplicationMessage   (* holt eine Nachricht mit bis zu 512 Byte Nutzdaten vom MMI ab *)
VAR_INPUT
    data_pab        : POINTER TO ARRAY[0..511] OF BYTE;
    datatypes_s     : STRING(255);
END_VAR
VAR_OUTPUT
    sb1_b           : BYTE;
    sb2_b           : BYTE;
    index_b         : BYTE;
    handle_b        : BYTE;
    len_w           : WORD;
    ok              : BOOL;
END_VAR
```

Parameter

data_pab	Zeiger auf empfangene Nutzdaten
datatypes_s	optionaler Datendescriptor, siehe DefDataTypes
sb1_b	Steuerblock 1 der Nachricht vom HMI
sb2_b	Steuerblock 2, optional
index_b	Blockzähler, optional
handle_b	Kennung für Absender, reserviert
len_w	Länge der empfangenen Nutzdaten in Bytes
ok	TRUE, wenn eine Nachricht empfangen wurde

Beschreibung

Mit dieser Funktion kann die SPS eine Nachricht vom Typ SB0_SPSAUFTRAG_KUC (SB0=14) mit bis zu 512 Byte Nutzdaten vom HMI abholen. Dabei wird der Nachrichtenpuffer zwischen NCR und MMI im Dual-Port-RAM verwendet. Die Funktion kann zusammen mit PutApplicationMessage verwendet werden, um eine schnelle Nachrichtenkommunikation zwischen SPS und HMI aufzubauen.

In *data_pab* kann ein Pointer auf einen Puffer übergeben werden, in dem die empfangene Nachricht abgelegt werden soll. Dabei ist sicherzustellen, dass dieser Puffer groß genug ist, um die empfangene Nachricht aufzunehmen. Beim Start oder Reset der SPS wird dieser Pointer automatisch auf den Anfang von DB7 (%MW7.0) initialisiert.

Wenn das Format der zu empfangenden Nachricht vorher bekannt ist, kann der Datendescriptor *datatypes_s* verwendet werden, um das eventuell notwendige Byte-Swapping der Nutzdaten automatisch vornehmen zu lassen. Andernfalls muss dies anschließend mit MOVESWAPPED durchgeführt oder die Funktionen GET_WORD, GET_DWORD, usw. verwendet werden, um die Daten aus der Nachricht zu holen. Sollte der angegebene Descriptorstring fehlerhaft sein, werden die Daten ohne Byte-Swapping übernommen.

Wenn beim Aufruf von GetApplicationMessage eine Nachricht vom HMI bereit steht, wird ok auf TRUE gesetzt und die Nachricht in den angegebenen Puffer kopiert. Die Funktion kehrt in jedem Fall sofort zurück, auch wenn keine Nachricht vom HMI bereit steht. In diesem Fall wird **ok** auf **FALSE** gesetzt.

Durch Auswertung von *sb1_b* kann die Nachricht von der SPS identifiziert werden. Die Festlegung der Steuerblöcke muss zwischen SPS- und MMI-Entwickler erfolgen. Die Länge der Nutzdaten wird in *len_w* in Bytes abgelegt.

Die Variablen *sb2_b* und *index_b* sind optional. Ihre Verwendung muss zwischen SPS- und HMI-Entwickler abgeklärt werden. *handle_b* ist für die gleichzeitige Kommunikation mit mehreren Applikationen reserviert.

Hinweis!

Die Übertragung dieser Nachrichten erfolgt grundsätzlich unsynchronisiert. In der Steuerung können bis zu 10 Nachrichten dieses Typs von MMI an SPS gepuffert werden, jede weitere geht verloren. In diesem Fall meldet die Steuerung eine entsprechende Fehlermeldung. Falls erforderlich müssen Synchronisationsmechanismen zwischen HMI und SPS vereinbart werden.

8.8.7.2 PutApplicationMessage

Deklaration

```
FUNCTION_BLOCK PutApplicationMessage   (* Sendet eine Nachricht mit bis zu 512 Byte
                                          Nutzdaten ans MMI *)
VAR_INPUT
    data_pab      : POINTER TO ARRAY[0..511] OF BYTE;
    datatypes_s   : STRING(255);
    sb1_b         : BYTE;
    sb2_b         : BYTE;
    index_b       : BYTE;
    handle_b      : BYTE;
    len_w         : WORD;
END_VAR
VAR_INPUT
    ok            : BOOL;
END_VAR
```

Parameter

data_pab	Zeiger auf zu sendende Nutzdaten
datatypes_s	optionaler Datendescriptor, siehe DefDataTypes
sb1_b	Steuerblock 1 der Nachricht ans HMI
sb2_b	Steuerblock 2, optional
index_b	Blockzähler, optional
handle_b	Kennung für Empfänger, reserviert
len_w	Länge der empfangenen Nutzdaten in Bytes
ok	TRUE, wenn eine Nachricht gesendet wurde

Beschreibung

Mit dieser Funktion kann die SPS eine Nachricht vom Typ SB0_SPSAUFTRAG_KUC (SB0=14) mit bis zu 512 Byte Nutzdaten ans HMI senden. Dabei wird der Nachrichtenpuffer zwischen NCR und MMI im Dual-Port-RAM verwendet. Die Funktion kann zusammen mit GetApplicationMessage verwendet werden, um eine schnelle Nachrichtenkommunikation zwischen SPS und MMI aufzubauen.

In *data_pab* kann ein Pointer auf einen Puffer mit den zu sendenden Nutzdaten übergeben werden. Beim Start oder Reset der SPS wird dieser Pointer automatisch auf den Anfang von DB6 (%MW6.0) initialisiert.

Der Datendescriptor *datatypes_s* kann verwendet werden, um das eventuell notwendige Byte-Swapping der Nutzdaten automatisch vornehmen zu lassen. Andernfalls muss dies vorher mit MOVESWAPPED durchgeführt oder die Funktionen PUT_WORD, PUT_DWORD, usw. verwendet werden, um die Daten in die Nachricht zu schreiben.

Wenn *datatypes_s* einen Leerstring enthält, muss die Anzahl der zu sendenden Nutzdatenbytes in *len_w* angegeben werden. Sonst ergibt sich die Länge aus dem Datendescriptor.

Mit *sb1_b* wird der Typ der Nachricht bestimmt. Für jedes verwendete Nutzdatenformat sollte ein eigener Steuerblock verwendet werden. Die Festlegung muss zwischen SPS- und MMI-Entwickler erfolgen.

Die Variablen *sb2_b* und *index_b* sind optional. Ihre Verwendung muss zwischen SPS- und HMI-Entwickler abgeklärt werden. *handle_b* ist für die gleichzeitige Kommunikation mit mehreren Applikationen reserviert.

Nach dem Eintragen der Nachricht in den Sendepuffer zum HMI, wird ok auf TRUE gesetzt. Bei einem fehlerhaften Descriptorstring *datatypes_s* oder einem Sendepufferüberlauf wird ok auf FALSE gesetzt. Die Funktion kehrt in jedem Fall sofort zurück, auch wenn die Nachricht nicht gesendet werden konnte. Im Falle eines Sendepufferüberlaufs kann der Aufruf natürlich später wiederholt werden.

Hinweis!

Die Übertragung dieser Nachrichten erfolgt grundsätzlich unsynchronisiert. In der Steuerung können insgesamt bis zu 30 Nachrichten von NCR an MMI gepuffert werden, jede weitere geht verloren. In diesem Fall meldet die Steuerung eine entsprechende Fehlermeldung. Falls erforderlich müssen Synchronisationsmechanismen zwischen HMI und SPS vereinbart werden.

8.8.8 Echtzeituhr (nur ETCxM)

8.8.8.1 RTC_GetTime_DT (nur ETCxM)

Deklaration

```
FUNCTION RTC_GetTime_DT: BOOL
VAR_INPUT
    GetTime_pr      : DWORD;
END_VAR
```

Beschreibung

Mit dieser Funktion kann die Echtzeituhr der Steuerung ausgelesen werden. Der Funktion muss die Adresse einer Variablen vom Typ DT übergeben werden (siehe Beispiel). Der Returnwert TRUE der Funktion signalisiert, dass die Zeit ausgelesen werden konnte.

Beispiel

```
ret_bit         : BOOL;
time_dt         : DT;

ret_bit = RTC_GetTime_DT(ADR(time_dt));
```

8.8.8.2 RTC_SetTime_DT (nur ETCxM)

Deklaration

```
FUNCTION RTC_SetTime_DT: BOOL
VAR_INPUT
    SetTime_dt      : DT;
END_VAR
```

Beschreibung

Mit dieser Funktion wird die Echtzeituhr der Steuerung auf den übergebenen Wert gesetzt. Der Returnwert TRUE der Funktion signalisiert, dass die Zeit in den Uhrenbaustein geschrieben wurde.

Beispiel

```
ret_bit         : BOOL;
time_dt         : DT := dt#2003-11-15-00:00:00;

ret_bit = RTC_SetTime_DT(time_dt);
```

8.9 Bibliothek ServerSDO.lib

8.9.1 InitServerSdo

Deklaration

```
FUNCTION InitServerSdo: BOOL
VAR_INPUT
    Max_NetVarODIdx     : UINT;
    pNetVarOD           : POINTER TO NetVarOD_CAN;
END_VAR
```

Parameter

MAX_NetVarODIdx	maximal Index
pNetVarOD	Zeiger auf das Objektverzeichnis

Beschreibung

Diese Funktion gibt dem Laufzeitsystem ein Objektverzeichnis zum Datenaustausch mit einem CANopen-Client bekannt.

Beispiel

Um das CoDeSys-Objektverzeichnis zu aktivieren, muss folgender Funktionsaufruf einmalig im SPS-Programm durchgeführt werden:

```
InitServerSdo(MAX_NetVarOD_CAN, ADR(pNetVarOD_CAN[0]));
```

9 Stichwortverzeichnis

A

Ablaufsteuerung, 62

AddCobIdCanMsg, 401

Adresse CAN, 56

Adressierung
- Datenbaustein, 343
- E/A-Modul, 342

Adressierung , 342

ALLOCV24, 369

Anhang, 314

Antriebe, parametrieren, 36

Antriebe überprüfen, mit ETC-MMI, 43 , 44

Antriebsregler, Kennzeichnung, 12

Automatisierungssystem, Beispiel, 13

B

Beispiel
- Automatisierungssystem, 13
- CNC-Programm, 47
- SPS-Programm, 58

Bestimmungsgemäße Verwendung, 12

Betriebsart MMI
- ”Automatik”, 292
- ”Diagnose”, 306
- ”Einrichten”, 287
- ”Programmieren”, 296

Betriebsarten ETCxH, 25

Betriebsdaten, 354

Bibliothek, 355

Bibliothek , ServerSDO.lib, 409

Bootloader, 357

Bootmonitor, Aufruf, 81

Busabschluss
- CAN-Bus, 15
- ME-Bus, 15

C

CAN
- Adresse, 56
- Busabschluss, 15
- Master konfigurieren, 337
- Module suchen, 341
- Slave konfigurieren, 338

CE-Konformität, 12

CenterString, 355

ClearCanMsg, 402

CloseCanMsg, 402

CLRRXBUFFER, 372

CLRTXBUFFER, 373

CNC-Programm
- eingeben, 48
- erweitern, 50
- Fehlermeldung, 72
- laden, 49
- speichern, 48
- starten, 49
- testen, 70

CNC-Programmierung, 45

CoDeSys, 330 , 333 , 356
- Adressierung , 342
  Datenbaustein, 343
  E/A-Modul, 342
- Betriebsdaten, 354
- Bibliothek, 355
- Bibliothek , ServerSDO.lib, 409
- CAN-Master konfigurieren, 337
- CAN-Slave konfigurieren, 338
- CANopen-Module suchen, 341
- Datenbausteine , 350
- E/A-Anzahl überprüfen, 340
- E/A-Module konfigurieren, 337
- Fehlermeldungen, 354
- Funktionen
  AddCobIdCanMsg, 401
  allgemein, 355
  ALLOCV24, 369
  CANopen-Funktionen, 400
  ClearCanMsg, 402
  CloseCanMsg, 402
  CLRRXBUFFER, 372
  CLRTXBUFFER, 373
  CopDefineDS403, 389
  CopGetNodeState, 391
  CopReadObjekt, 392
  CopWriteObject, 394
  CopXDefineDS403, 391
  CopXDisableSync, 398
  CopXEnableSync, 397
  CopXReadObject, 393
  CopXSetOperational, 396
  CopXSetPreoperational , 397
  CopXWriteObject, 395
  CopyChannelDisplayData, 398
  CopyCyclicDisplayData, 399
  CopyDiagDisplayData, 399

DEFDATATYPES, 384
DefineCanMsg, 400
DelCobIdCanMsg, 401
DrvReadObject, 392
DrvWriteObject, 394
FMOD, 355
Format, 356
FREEV24, 371
Gerätetreiber, 381
GET_BYTE, 386
GET_DINT, 386
GET_WORD, 386
GET_INT, 386
GET_LREAL, 386
GET_REAL, 386
GetApplicationMessage, 405
GetFirmwareVersion, 357
GetMacAddr, 357
GetUserParam, 358
Konstanten V24-Funktionen, 373 , 381
INITV24, 370
IO_SET, 358
LOAD, 369 , 374
LOAD_PARAM, 369
Load_Param, 359
MEMCOMP, 386
MEMCOPY, 386
MEMSET, 387
OVESWAPPED, 387
PUT_BYTE, 388
PUT_DWORD, 388
PUT_INT, 388
PUT_LREAL, 388
PUT_DINT, 388
PUT_WORD, 388
PutApplicationMessage, 406
READ_PARAM_DINT, 359
READ_PARAM_INT, 359
READ_PARAM_LREAL, 360
READ_PARAM_REAL, 360
READ_SYSPARAM, 360
READ_TOOLDATA, 360
READBLOCKV24, 371
ReadCanMsg, 403
READV24, 371
RTC_GetTime_DT, 408
RTC_SetTime_DT, 408
SAVE, 369 , 375
SAVE_PARAM, 361 , 369
CenterString, 355
SetCurrentPath, 375
SETINPUT_BIT, 362
SETINPUT_WORD, 362
SETLANGUAGE, 365
SINGLEBLOCK, 362
SPSERROR, 363
STRTOF, 365
STRTOL10, 365
SYSCLOSEFILE, 376
SYSDISKFORMAT, 378
SYSDISKINFO, 380
SYSERROR, 366
SYSFIRSTFILE, 379
SYSNEXTFILE, 379
SYSOPENFILE, 375
SYSREADFILE, 376
SYSREADLINE, 377
SYSREMOVEFILE, 378
SYSWRITEFILE, 377
TRACE, 366
WordWrap, 367
WRITE_PARAM_DINT, 367
WRITE_PARAM_INT, 367
WRITE_PARAM_LREAL, 368
WRITE_PARAM_REAL, 367
WRITE_SYSPARAM, 368
WRITEBLOCKV24, 372
WriteCanMsg, 403
WRITEV24, 372
- installieren, 330
- Maschinenkonstanten, 353
- Netzwerkvariablen, 346
- Objektverzeichnis, 343
- Parameterfeld, 353
- Parametermanager, 343
- Variablenliste, global, 345 , 346 , 349
- programmieren mit, 51 , 329
- Projekt anlegen, 52
- Remanente Variablen, 343
- Schnittstelle zur ETC, 350
- SPS-Tasks konfigurieren, 333 , 335
- starten, 52
- Steuerung konfigurieren, 51
- Systemvariablen, 350 , 352
- Zielsystem einstellen, 333
- Zielsystemeinstellungen, 346

CopDefineDS403, 389

CopGetNodeState, 391

CopReadObjekt, 392

CopWriteObject, 394

CopXDefineDS403, 391

CopXDisableSync, 398

CopXEnableSync, 397

CopXReadObject, 393

CopXSetOperational, 396

CopXSetPreoperational , 397

CopXWriteObject, 395

CopyChannelDisplayData, 398

CopyCyclicDisplayData, 399

CopyDiagDisplayData, 399

D

Datei
- EDS-, 55
- ETCxC.mk, 39

Dateimanager, 302

Datenbaustein, 60

Datenbaustein 0, 239

Datenbaustein 1, 242

Datenbaustein 15, 267

Datenbaustein 2, 262

Datenbaustein 8 ... 14, 264

Datenbausteine , 350

Datenfelder
- P-Feld, 178
- Q-Feld, 187

DEFDATATYPES, 384

DefineCanMsg, 400

DelCobIdCanMsg, 401

DPR-Schnittstelle, 332

DrvReadObject, 392

DrvWriteObject, 394

E

E/A-Anzahl überprüfen, 340

E/A-Module konfigurieren, 63 , 337

Editor, 47 , 299

EDS-Datei, 55

Ein-/Ausgänge definieren, 63

Erste Schritte, 13

ETC und PC verbinden, 331

ETC-Insel, 15

Datei, DelphMMI.ini, 319

ETC-MMI
- Antriebe überprüfen, 43 , 44
- bedienen, 283 , 284
- Beschreibung, 280
- Betriebsart ”Automatik”, 292
- Betriebsart ”Diagnose”, 306
- Betriebsart ”Einrichten”, 287
- Betriebsart ”Programmieren”, 296
- Bildaufteilung, 283
- Dateimanager, 302
- DELPHMMI.INI, 319
- Dialogbox zur Dateiauswahl, 300
- Editor, 47 , 299
- Hilfefunktion, 285
- installieren, 31 , 280
- Konfigurationsdatei, 319
- konfigurieren, 30
- Maschinenkonstanten anpassen, 42
- Maschinenkonstanten überprüfen, 43 , 44
- Meldungen, 286
- Passwort, 286
- Sprachdatei, 314
- Sprache umschalten, 33
- SPS-Tasten, 73
- starten, 32 , 282
- Verbindung herstellen, 34
- Zyklenprogrammierung, 303 , 318

ETC-MMI-Gateway
- Beschreibung, 270
- Fehlerlogbuch, 275 , 277
- Funktionsweise , 271
- installieren, 270
- Kommunikationswerte im DPR Bereich, 279
- Konfigurationsoberfläche, 272
- konfigurieren, 30 , 272
- konfigurierte Verbindungen, 272
- Mmigtway.ini, 277
- starten, 271
- Verbindung herstellen, 34
- Versionsinformationen, 276

ETC-Programmierung
- Datenfelder, 178 , 187
- Formelprozessor , 170
- G-Funktionen, 90
- H-Funktionen, 173
- M-Funktionen, 173
- Modale Funktionen, 88
- Parameterzuweisung P, 172
- Programm übertragen, 85
- Programm-Komponenten, 85
- Q-Funktionen, 175
- Rechenoperationen, 170
- Satzerweiterungen, 172
- Satzvorverarbeitung, 89
- S-Funktionen, 176 , 177
- Varianten der Programm-Erstellung, 85
- Wegbedingung G, 87
- Zeitsynchronisation, 89

ETC-Schnittstelle SPS-NC
- Datenbaustein 0, 239
- Datenbaustein 1, 242
- Datenbaustein 15, 267
- Datenbaustein 2, 262
- Datenbaustein 8 ... 14, 264

ETCHx
- Anschlüsse, 15
- Beschreibung, 13
- Betriebsart, 25
- IP-Adresse, 28
- mit PC verbinden, 16
- starten, 21

ETCPx
- Beschreibung, 13
- installieren, 31

ETCxC.mk, 39

Ethernet-Schnittstelle, 331

F

Fehlermeldungen, 354

Firmware, Update, 81

FMOD, 355

Format, 356

Formelprozessor , 170

FREEV24, 371

Funktionen
- AddCobIdCanMsg, 401
- allgemein, 355
- ALLOCV24, 369
- CANopen-Funktionen, 400
- ClearCanMsg, 402
- CloseCanMsg, 402
- CLRRXBUFFER, 372
- CLRTXBUFFER, 373
- CopDefineDS403, 389
- CopGetNodeState, 391
- CopReadObjekt, 392
- CopWriteObject, 394
- CopXDefineDS403, 391
- CopXDisableSync, 398
- CopXEnableSync, 397
- CopXReadObject, 393
- CopXSetOperational, 396
- CopXSetPreoperational , 397
- CopXWriteObject, 395
- CopyChannelDisplayData, 398
- CopyCyclicDisplayData, 399
- CopyDiagDisplayData, 399
- DEFDATATYPES, 384
- DefineCanMsg, 400
- DelCobIdCanMsg, 401
- DrvReadObject, 392
- DrvWriteObject, 394
- FMOD, 355
- Format, 356
- FREEV24, 371
- Gerätetreiber, 381
- GET_BYTE, 386
- GET_DINT, 386
- GET_WORD, 386
- GET_INT, 386
- GET_LREAL, 386
- GET_REAL, 386
- GetApplicationMessage, 405
- GetFirmwareVersion, 357
- GetMacAddr, 357
- GetUserParam, 358
- I/O-Funktionen, 381
- INITV24, 370
- IO_SET, 358
- Konstanten V24-Funktionen, 373
- LOAD, 369 , 374
- LOAD_PARAM, 369
- Load_Param, 359
- MEMCOMP, 386
- MEMCOPY, 386
- MEMSET, 387
- OVESWAPPED, 387
- PUT_BYTE, 388
- PUT_DINT, 388
- PUT_DWORD, 388
- PUT_INT, 388
- PUT_REAL, 388
- PUT_WORD, 388
- PutApplicationMessage, 406
- READ_PARAM_DINT, 359
- READ_PARAM_INT, 359
- READ_PARAM_LREAL, 360
- READ_PARAM_REAL, 360
- READ_SYSPARAM, 360
- READ_TOOLDATA, 360
- READBLOCKV24, 371
- ReadCanMsg, 403
- READV24, 371
- RTC_GetTime_DT, 408
- RTC_SetTime_DT, 408
- SAVE, 369 , 375
- SAVE_PARAM, 361 , 369
- CenterString, 355
- SetCurrentPath, 375
- SETINPUT_BIT, 362
- SETINPUT_WORD, 362
- SETLANGUAGE, 365
- SINGLEBLOCK, 362

- SPSERROR, 363
- STRTOF, 365
- STRTOL10, 365
- SYSCLOSEFILE, 376
- SYSDISKFORMAT, 378
- SYSDISKINFO, 380
- SYSERROR, 366
- SYSFIRSTFILE, 379
- SYSNEXTFILE, 379
- SYSOPENFILE, 375
- SYSREADFILE, 376
- SYSREADLINE, 377
- SYSREMOVEFILE, 378
- SYSWRITEFILE, 377
- TRACE, 366
- WordWrap, 367
- WRITE_PARAM_DINT, 367
- WRITE_PARAM_INT, 367
- WRITE_PARAM_LREAL, 368
- WRITE_PARAM_REAL, 367
- WRITE_SYSPARAM, 368
- WRITEBLOCKV24, 372
- WriteCanMsg, 403
- WRITEV24, 372

G

G-Funktion
- Einzelbeschreibungen, 93
- Grundlagen, 45
- Übersicht, 90

G00 Lineare Zustellung im Eilgang, 93

G01 Geradeninterpolation, 95

G02 Kreisinterpolation, 97

G04 Verweilzeit, 100

G05 Spline - Interpolation, 101

G06 Polynom - Interpolation, 103

G100 Polarkoordinaten: Geradeninterpolation im Eilgang, 124

G101 Polarkoordinaten: Geradeninterpolation, 125

G102/G103 Polarkoordinaten: Kreisinterpolation, 126

G110 Polarkoordinaten: Mittelpunkt übernehmen, 127

G111 Polarkoordinaten: Mittelpunktprogrammierung, 127

G112 Tangentiale Nachführung ein, 128

G113 Tangentiale Nachführung aus, 129

G114 6-Achsen-Transformation, 129

G115 Mantel-Transformation, 130

G116 Rotationsachsen-Transformation, 131

G120 Abs. Koordinatenverschiebung über modalen Offset, 132

G121 Programmieren des modalen Offsets, 133

G122 Konfigurieren der Wirkung der Verfahrtasten, 134

G125 Parameterfeldvergleich, 135

G130 Parameterfeldvergleich, 136

G131 Modale Vergleichsoperation löschen, 138

G132 Modale Programmverzweigung sperren, 138

G133 Modale Programmverzweigung erlauben, 138

G134 Parameterfeldvergleich, 139

G140/G141/G142 Unrundschleifen, 140

G143 Parameter von Schleifphasen beim Unrundschleifen, 142

G144/G145 Programmieren einer Korrekturtabelle beim Unrundschleifen, 143

G150 Q-Bit Vergleich, 144

G151 Q-Bit Vergleich, 146

G152 Q-Bit Vergleich, 147

G153 Warten auf die Terminierung eines Kanal 1, 148

G158/G159 Intermittierender Betrieb, 149

G16 Auswahl der Haupt- und Nebenachsen, 105

G161 Istpositionsübernahme, 150

G162 Achsverbund definieren, 151

G17, G18, G19 Ebenenauswahl, 106

G175 Achstausch, 152

G180 Modales Fahren einschalten, 153

G181 Modales Fahren ausschalten, 154

G187 Handrad, 154

G193 Absolutes Nullpunktsetzen, 155

G194 Programmieren eines zus. Grundoffsets, 155

G195 Absolute Koordinatenverschiebung , 156

G20 Satzsprung, 106

G200 Ein-/Ausschalten des Geometriefilters, 157

G201 Ändern der Beschleunigungs- und Bremsrampen, 158

G209 Setzen des Geometriezählers, 159

G211 Transformation für zweiachsige Knickarmroboterkinematik, 159

G22 Unterprogrammaufruf, 107

G226 Hardwareendschalter umkonfigurieren, 161

G233 2D/3D-Achsenkorrektur, 162

G234 Richtungsabhängige Höhenkorrektur, 164

G24 Pos. Verfahrbereichsgrenze definieren, 108
G25 Neg. Verfahrbereichsgrenze definieren, 109
G251 Sprungantwort einer Achse aufnehmen, 165
G252 Eingabe eines Wertes über Anzeigegerät, 166
G253 Ausgabe eines Kommentars, 168
G26 Verfahrbereichsgrenzen aktivieren, 109
G27 Sprungfunktion mit Wiederholungszähler, 110
G30, G31 Strecken- bzw. Bahnsteuerung, 110
G33 Kopplung einschalten, 111
G34 Kopplung ausschalten, 112
G36 Modales Oszillieren einschalten, 112
G37 Modales Oszillieren ausschalten, 113
G40 Aufheben der Werkzeugkorrektur, 113
G41, G42 Werkzeugbahnkorrektur, 114
G53 Temp. Koordinatenverschiebung aufheben, 115
G54 Temporäre Nullpunktverschiebung, 115
G60 Genauhalt Ein/Aus, 116
G61 Satzvorverarbeitung anhalten, 117
G74 Referenzpunktfahrt, 117
G75 Maßstabsfaktoren für Eingabeeinheiten, 118
G76 Maßstabsfaktoren für Impulsbewertung, 118
G88 Grunddrehung, 119
G89 Konturdrehung, 120
G90 Absolute Maßangaben (Bezugsmaß), 121
G91 Inkrem. Maßangaben (Kettenmaßprogr.), 121
G92 Relative Nullpunktverschiebung, 122
G93 Absolute Nullpunktverschiebung, 123
G99 Rücksprung, 123
Gerätetreiber, 381
GET_BYTE, 386
GET_DINT, 386
GET_DWORD, 386
GET_INT, 386
GET_LREAL, 386
GET_REAL, 386
GET_WORD, 386
GetApplicationMessage, 405
GetFirmwareVersion, 357
GetMacAddr, 357
GetUserParam, 358
Gewährleistung, 12
Guard Time, 57

H

H-Funktionen, 173
Haftung, 12
Hersteller, 12
HMI, 75
HyperTerminal, 21

I

Inbetriebnahme, Inbetriebnahmeschritte, Übersicht, 20
INITV24, 370
IO_SET, 358
IP-Adresse, ETCHx, 28
IPC
- Applikationen, 14
- Beschreibung, 14

K

Kennzeichnung, Antriebsregler, 12
Konformität, 12
Konstanten
- CANopen-Funktionen, 400
- I/O-Funktionen, 381
- V24-Funktionen, 373

L

LENZE-HMI, 75
Life Time Factor, 57
LOAD, 369 , 374
LOAD_PARAM, 369
Load_Param, 359

M

M-Funktion, Grundlagen, 46
M-Funktionen, 173
MAC-Adresse, 29

Maschinenkonstanten, 94 , 110 , 158 , 353
- Datei anpassen, 41 , 42
- Datei in Steuerung laden, 40
- ETCxC.mk, 39
- Konfiguration der Achsen, Achskorrektur, 213
- Konfiguration der Achsen, Arbeitsfeld, 208
- Konfiguration der Achsen, Auflösung, 207
- Konfiguration der Achsen, Geschwindigkeit und Beschleunigung, 211
- Konfiguration der Achsen, Grundlegendes, 202
- Konfiguration der Achsen, Handräder, 215
- Konfiguration der Achsen, Referenzierung, 210
- Konfiguration der Achsen, Reglereinstellung, 209
- Konfiguration der Achsen, Synchronachsen, 217
- Konfiguration der Achsen, Zuordnung und Auswertung, 205
- Software-Konfiguration , 191
- Speicherplatzreservierung, 198
- Technologiespezifische Einstellungen, 218
- Testeinstellungen, 190
- überprüfen, 43 , 44
- Übersicht, 36 , 223

ME-Busabschluss, 15

MEMCOMP, 386

MEMCOPY, 386

MEMSET, 387

MK_ACHSENART, 206 , 215 , 217

MK_APPLACHSIDX, 203 , 205 , 215

MK_BAHNBESCHL, 212

MK_BAHNBREMS, 212

MK_BESCHL, 211

MK_BREMS, 211

MK_CANDRIVES, 202 , 205 , 215

MK_CANOPEN_BAUDRATE, 196

MK_CONST_REL_INCH, 193

MK_CONST_REL_MM, 193

MK_DELTAT, 196

MK_DW224_255, 222

MK_EPSILONGRAD, 195

MK_EPSILONMM, 195

MK_FEHLERRESTART, 195

MK_GENAUHALTZEIT , 208

MK_GEWINDE_VMAX, 221

MK_GRUNDOFFSET, 208

MK_HANDRADFAKTOR, 216

MK_HANDRADFILTER, 216

MK_HANDRADZUORDNUNG, 216

MK_HEADERANZAHL, 199

MK_IMPULSE, 207 , 215

MK_KANALANZAHL, 199

MK_KARTESISCH_ACHSNR, 219

MK_KONTURFEHLER, 193

MK_KUNDE, 191

MK_LAH_GRENZWINKEL, 194

MK_LAH_RUECKLAUFGRENZE, 200

MK_LAH_VORLAUFTIEFE, 200

MK_MASCH_POLAR_KART, 219

MK_MASSSTAB, 207

MK_METRISCH, 192

MK_MFKT_UPR_TABELLE, 218

MK_MODVMAX, 211

MK_NCPROG_NICHT_INS_EEPROM, 192

MK_NCPROG_OHNE_KOMMENTARE, 192

MK_OVERRIDEMAX, 195

MK_PFELD_GROESSE, 201

MK_POLAR_ACHSNR, 220

MK_RADIUS_B_BEWERTUNG, 194

MK_REF_RICHTUNG_UND_FOLGE, 210

MK_S_VERSATZSPERRE, 196

MK_S0T0_VERSATZ_ERLAUBT, 196

MK_SCHLEPPGENAUHALT, 208

MK_SPINDELUMKEHRSPIEL, 213

MK_SPS_DATENGROESSE, 198

MK_SPS_DUMMY, 190

MK_SPS_SPEICHERGROESSE, 198

MK_SPV_SPEICHERGROESSE, 198

MK_SPV_SYMBOLANZAHL, 199

MK_SW_ENDS_MINUS, 208

MK_SW_ENDS_MIT_RAMPE, 208

MK_SW_ENDS_PLUS, 208

MK_SYNCHRONABWEICHUNG, 217

MK_SYNCHRONOFFSET , 217

MK_T_BAHNBESCHL, 212

MK_T_BESCHL , 211

MK_T2, 209

MK_TECHNOLOGIEDATEN, 218

MK_TEST_OHNEMECHANIK, 190

MK_VBAHNMAX, 212
MK_VMAX, 211
MK_VOREINSTELLUNG, 191
MK_WEG, 207 , 215
MK_WLK_C_GRENZWINKEL, 220
MK_WLK_C_OFFSET, 220
MK_WLK_VERWEILZEIT, 221
MK_X_WINKEL, 221
Mmigtway.ini, 277
Monitor-Schnittstelle
- aktivieren, 21
- Befehle, 23
- Störungsbeseitigung, 24

MUSTER.MK, 223

N

Netzwerkvariablen, 346
Nodeguarding, 57 , 339 , 341

O

Objektverzeichnis, 343
OVESWAPPED, 387

P

P-Feld, 178
Parameter
- Achs-spezifische, 179
- Programmverwaltung, 181
- System, 179
- Technologie-spezifische, 184
- Technologie-spezifische Anwenderparameter, 185
- Zeiterfassung, 185

Parameterfeld, 353
Parametermanager, 343
Parameterzuweisung P, 172
Parametrieren, Antriebe, 36
PC
- Applikationen, 14
- Beschreibung, 14

PC und ETC verbinden, 331
Produktkennung
- Module ETCHx, 11
- Steuerung ETCHx, 11
- Steuerung ETCPx, 11

Programm generieren, 349
Programmierung mit CoDeSys, 51 , 329
PUT_BYTE, 388
PUT_DINT, 388
PUT_DWORD, 388
PUT_INT, 388
PUT_LREAL, 388
PUT_REAL, 388
PUT_WORD, 388
PutApplicationMessage, 406

Q

Q-Feld, 187
Q-Funktionen, 175

R

READ_PARAM_DINT, 359
READ_PARAM_INT, 359
READ_PARAM_LREAL, 360
READ_PARAM_REAL, 360
READ_SYSPARAM, 360
READ_TOOLDATA, 360
READBLOCKV24, 371
ReadCanMsg, 403
READV24, 371
Rechenoperationen, 170
Rechtliche Bestimmungen, 12
Remanente Variablen, 343
RTC_GetTime_DT, 408
RTC_SetTime_DT, 408

S

S-Funktionen, 176
Satzerweiterungen, 172
Satzvorverarbeitung, 89
SAVE, 369 , 375
SAVE_PARAM, 361 , 369
Schnittstelle
- DPR, 332
- Ethernet, 331
- Monitor-, 21 , 23
- V.24, 331

Schnittstelle zur ETC, 350
ServerSDO.lib, 409

SetCurrentPath, 375

SETINPUT_BIT, 362

SETINPUT_WORD, 362

SETLANGUAGE, 365

SINGLEBLOCK, 362

Sprache umschalten, ETC-MMI, 33

SPRACHE.TXT, 314

SPS-Programm
- Ablaufsteuerung, 62
- bedienen, 69
- Beispiel, 58
- Bibliotheken einfügen, 59
- erstellen, 58
- konfigurieren, 59
- laden, 66 , 69
- starten, 59 , 69
- testen, 70

SPS-Tasks konfigurieren, 333 , 335

SPS-Tasten, 73

SPSERROR, 363

Statusmeldungen, 17

Steuerung konfigurieren, 51

Störungsbeseitigung, Monitor-Schnittstelle, 24

STRTOF, 365

STRTOL10, 365

SYSCLOSEFILE, 376

SYSDISKFORMAT, 378

SYSDISKINFO, 380

SYSERROR, 366

SYSFIRSTFILE, 379

SYSNEXTFILE, 379

SYSOPENFILE, 375

SYSREADFILE, 376

SYSREADLINE, 377

SYSREMOVEFILE, 378

Systemvariablen, 60 , 350 , 352

SYSWRITEFILE, 377

T

T-Funktionen, 177

Terminalprogramm, 21

Testen
- CNC-Programm, 70
- SPS-Programm, 70

TRACE, 366

Typenschild
- Module ETCHx, 11
- Steuerung ETCHx, 11
- Steuerung ETCPx, 11

U

Übersicht
- G-Funktion, 90
- Maschinenkonstanten, 36 , 223
- Rechenoperationen, 170

Update, Firmware, 81

V

V.24-Schnittstelle, 331

Variablenliste, global, 345 , 346

Verwendung, bestimmungsgemäße, 12

W

Wegbedingung G, 87

WordWrap, 367

WRITE_PARAM_DINT, 367

WRITE_PARAM_INT, 367

WRITE_PARAM_LREAL, 368

WRITE_PARAM_REAL, 367

WRITE_SYSPARAM, 368

WRITEBLOCKV24, 372

WriteCanMsg, 403

WRITEV24, 372

Z

Modale Funktionen, 88 , 89

Zielsystem einstellen, 333

Zielsystemeinstellungen, 346

Zyklenprogrammierung, 303 , 318

Lenze Drive Systems GmbH
Hans-Lenze-Straße 1
D-31855 Aerzen
Germany

☎ +49 (0) 51 54 82-0
☎ Service 00 80 00 24 4 68 77 (24 h helpline)
🖷 Service +49 (0) 51 54 82-1112
E-Mail Lenze@Lenze.de
Internet www.Lenze.com

EDSTCXN
DE 2.0
© 07/2006
TD29

10 9 8 7 6 5 4 3 2 1